Optimising the Output of an Articulated WEC Through the Use of a Biologically-Inspired and Artificially Evolved Neural Network

Timothy R Mundon

A thesis submitted for the degree of Doctor of Philosophy.
The University of Edinburgh.
May 2, 2006

Abstract

This thesis represents work that investigates the implementation of a practical control strategy for a simple wave energy converter in regular and irregular waves. Inspired by the neurophysiology of the lamprey, this work examines the limitations of using this neural structure as a method for control of an articulated wave energy converter.

Starting with a very simple mechanical model of a single heaving buoy, evolutionary techniques are employed to evolve a single lamprey segment that will be capable of acting as a controller. Without prediction of the incident wave, wide-bandwidth latching controllers are evolved that show significant improvements over optimal real damping in increasingly complex waveforms.

More complex mechanical configurations are also investigated, expanding the simple heaving buoy into two and three interconnected buoys with power being developed through their relative motion. Neural latching controllers are evolved for these different configurations and it is shown that an effective neural latching controller cannot be evolved for more than two interconnected buoys.

This thesis investigates the processes of producing a viable implementation of a latching strategy using neural oscillators as a non-linear feedback loop. It covers their performance in regular and irregular waves and demonstrates the limitations of latching control when applied to an articulated system.

Declaration of Originality

I hereby declare that the research recorded in this thesis and the thesis itself was composed and originated entirely by myself in the School of Engineering and Electronics at The University of Edinburgh.

Tim Mundon

Acknowledgements

I would like to thank the following people, in no particular order, for their contribution to this work.

- Alan Murray, John Hallam and Robin Wallace, my supervisors at Edinburgh University, for their unquestionable contribution towards this project.
- Leena Patel, for the original lamprey C code.
- Dave Pizer and Richard Yemm at OPD for initial guidance.
- Johannes Falnes, for offering help and advice on my first paper.
- Roy and Margaret Mundon, my parents, for providing all kinds of support.
- Finally I would like to thank my partner, Megan Lickliter for her unending support, attention and proof-reading. None of this would be complete without you.

Contents

List of fi gures

List of tables

Glossary

Added damping	Describes the force resulting from the oscillations emanating from an oscillating body in a fluid. Also known as Radiation Resistance.
Added Mass	Describes the mass of water a moving body 'drags' along with it.
Caudal	'Tail' end of the spinal cord.
Causal	Describes a type of control strategy that acts in a cause and effect manner upon the input wave with no future knowledge of the wave elevation. Conversely a "non-causal" strategy describes a condition whereby the strategy does have a knowledge of the future wave.
CCIN	Contralaterally and Caudally projecting Interneuron.
Chunk	Single global time-step.
Chunksize	This refers the the global time-step used in the system simulator.
Contra-lateral	Describes the symmetrical synaptic connections that stretch from one side of the CPG segment to the other.
Controller	Describes a neural 'segment' derived from the lamprey CPG that is evolved so as to effectively apply latching control to a single damper, C_{PTO}.
CPG	Central Pattern Generator.
EC	Edge Cell(sensory input).
EIN	Excitatory Interneuron.
GA	See Genetic Algorithm.
Genetic Algorithm	Optimisation routine, based upon genetic operators such as mating, selection and mutation.
Inter-segmental	Synaptic connections that propagate between segments.
Intra-segmental	Synaptic connections that are wholly within a segment.
LIN	Lateral Interneuron.
MFLOPS	Million floating point operations per second.
Neural Network	Computational network of individual processing elements (neurons) whose functionality is a result of the interconnections between neurons.
Neuronal Network	Collection of electrochemical neurons that can commonly be found in biology. Can be modelled in the same way as computational neural networks but requires a more complex neuron model. *Note: The terms neural and neuronal are used interchangeably in this document.*
MN	Motor Neuron.
Non-causal	See Causal.
Optimal Real Damping	An uncontrolled method that uses a fixed, optimum value for the power take-off damper to produce power in a WEC.
Pelamis	Articulated WEC device produced by Ocean Power Delivery.
Radiation resistance	See added damping.
Rostral	'Head' end of the spinal cord.
Runge-Kutta	Method for solving ordinary differential equations.
WEC	Wave Energy Converter.

List of Symbols

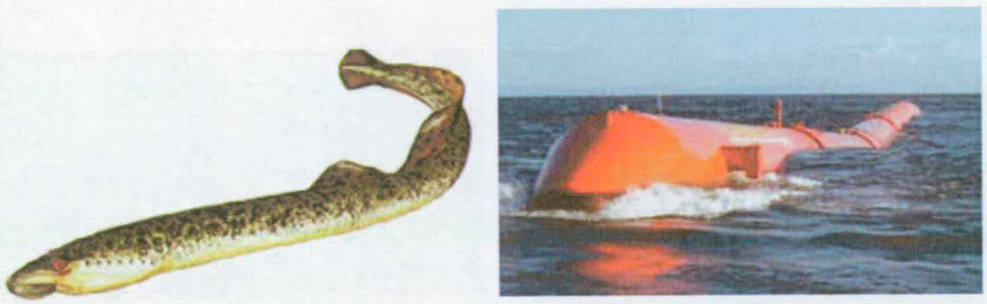
α	Constant for wave spectrum.
γ	Constant for wave spectrum.
ω	Angular frequency.
ω_0	Resonant frequency.
λ	Wavelength.
η	Wave elevation.
ρ/ρ_w	Density.
Θ	Neuron threshold.
Γ	Neuron gain.
τ_D	Neuron time constant.
τ_A	Time constant of neuron frequency adaptation.
μ	Coefficient of neuron frequency adaptation.
$\Xi_{+/-}$	Pre synaptic neuron.
$\Psi_{+/-}$	Neuron population.
c	Velocity.
C	Damping.
C_a	Radiation damping.
C_{PTO}	Power take-off damping. Analogous to $C_{PTO(latch-off)}$.
$C_{PTO(latch-on)}$	Power take-off damping for the system when buoy is latched (fixed).
$C_{PTO(latch-off)}$	See C_{PTO}.
E_t	Total energy content.
E_p	Potential energy content.
E_k	Kinetic energy content.
f	Frequency.
F_b	Total buoyancy force on buoy.
F_e	Force on buoy due to wave.
g	Acceleration due to gravity.
H_{RMS}	RMS wave elevation.
k	Wavenumber or angular repetancy.
k	Spring constant. (buoyancy)
M	Mass.
M_a	Added mass.
$S_{(\omega)}$	Wave Spectra.
T	Wave period.
u	Mean neuron firing frequency.
v_p	Phase velocity.
v_g	Group velocity.
$z, \dot{z}, \ddot{z}$	Buoy displacement, velocity and acceleration.

Chapter 1
Introduction

1.1 Overview

This document presents research that is a combination of work drawn from two very separate areas, marine energy extraction and artificial neurophysiology. These fields are combined within this project to develop a biologically inspired control method that will effect a practical control method for a simple wave energy converter.

Inspired by Ijspeert's work on artificially modelling the lamprey biological swimming controller, this thesis seeks to use the main characteristics of this artificial network in a different arena. The lamprey's locomotion is generated by propagating a travelling wave along the length of its body and it can be seen (see figure 1.1) that certain wave energy converters use a similar *anguiliform* motion to generate power.



(a) The lamprey. **(b)** The Pelamis WEC.

Figure 1.1: *The similarities in motion between the lamprey and PELAMIS.*

The lamprey's locomotive behaviour is a combination of muscle actuation and sensory feedback and is generated within a separate neural construct known as the central pattern generator. This mechanism allows the lamprey to swim without requiring direct control by the brain of the body movements and it is thought that, as the Pelamis shares a similar body motion, it may benefit from a similar method of control. Similarities between the two can be seen at a deeper level as

both devices share a segmental structure; where the Pelamis consists of a series of articulated units that generate power, the lamprey consists of a series of interconnected neural oscillators that control movement. Of course the scales are very different with only four segments in the Pelamis but over one hundred in the lamprey, although we can show that the systems are similar but the direction of control is slightly different; whereas the lamprey ***moves*** by propagating a wave along the length of its body, the Pelamis ***responds to*** a wave propagating along the length of its body.

The awesome power available in the marine environment is clear to anyone who has ever set foot on a boat or witnessed waves crashing into the shore on a windy day. The only issue faced is how to efficiently capture this power and turn it into useful electricity. This is a notoriously difficult problem, and although research has been concentrated on finding a viable method of conversion for over thirty years, there are still no commercial wave-power installations[1]. One of the primary reasons for this fact is the very high energy density present in ocean waves, meaning any device operating in this environment must be designed to withstand the extremes, yet generate power at a good rate of economic return.

The Pelamis wave energy converter (WEC) is an example of an articulated device designed to capture energy from ocean waves. It comprises of four sections that are linked via hydraulics, with the wave induced relative motion of these segments pumping hydraulic fluid through an accumulator to a generator. This or any other type of WEC requires that this wave induced motion is resisted by a damper (which could be the resistance applied by the electrical generator or possibly through some other means) but, if this damping can be adjusted at specific times in the wave cycle, we can maximise the power developed. This is the principle behind control of wave energy devices and, in ideal situations, can lead to very significant power increases. By using the analogy between the lamprey and the Pelamis as a starting point, it is thought that we can use the neural swimming controller of the lamprey to control the application of damping to a wave energy converter.

The Pelamis is a particularly complex mechanical system and as this level of complexity may not be entirely suitable for this thesis, a simplified model is developed that will allow us to start with a very basic system and then expand into an approximation of an articulated device. Starting with a single heaving buoy we add additional heaving buoys to the system and by using

[1] In summer 2005, Ocean Power Delivery has agreed to sell three of its Pelamis devices to Portugal to begin the first commercial wave-farm, but currently there are no commercial schemes operating.

their relative motions, the angular movements observed at the joints of an articulated device can be modelled.

This simplified mechanical system is modelled numerically and it can be seen[2] that this provides an acceptable alternative to a physical model. Whilst physical models can provide a much clearer impression of a device's performance than numerical models, numerical models offer an invaluable tool for initial development of WEC's and other marine structures.

Construction of an accurate physical model can be complex and wave tank facilities can be expensive. As a result, a significant amount of conceptual and evaluation work on WEC's is done with numerical models. The development of the Pelamis has involved a great deal of numerical modelling, following the typical use as an initial pre-cursor to device testing, the physical model then provides a calibration and verification of the numerical model[5].

The lamprey is an eel-like fish and features a simple and very well documented neural structure. Its locomotion, like all vertebrates, is centrally controlled by a neural structure within the spinal cord known as the central pattern generator (CPG). This neural structure extends along the entire length of the body and consists of a series of interconnected segments that directly control the muscles on each side of the body. It can be seen that although each segment can operate in isolation as an oscillator, when connected together, they feature precise time-delays that ensure the lamprey can move at an optimum velocity.

Due to this simple neural structure the lamprey has been seen as a perfect candidate for further study. Most particularly relevant is the work done by Ekeberg on modelling the lamprey's central pattern generator [6] and then Ijspeert who took Ekeberg's models and developed artificial central pattern generators using evolutionary techniques. This thesis takes Ijspeert's basic method and attempts to develop artificial neural controllers based upon the topology of the lamprey that are suitable for controlling a wave energy converter.

1.1.1 Development

The primary aim of this project was to discover whether the unique characteristics of the lamprey CPG can be used to apply control to a simple wave energy converter and further to see if this neural mechanism may be applied to a more complex articulated WEC.

[2]It can be seen that a number of important papers on wave energy use numerical modelling exclusively.[1–4]

A numerical model has been developed that allows the simulation of a single degree of freedom heaving buoy, restrained by a tether connected to the sea floor which incorporates a power take-off element. Basic hydrodynamic and hydrostatic parameters are included in the model, providing an adequate approximation to a real device. In addition to this numerical WEC model, a system was also developed (with the help of Leena Patel) that models the operation of the lamprey neuronal network. This model uses the equations developed by Ijspeert to simulate the operation of individual neural segments.

These neural and mechanical systems are developed such that they may be combined, allowing the neural network to influence the mechanical system and also the mechanical system to effect the operation of the neural network. This simulator forms the basis for all the experiments presented in this thesis and allows for different WEC and neural configurations to be tested in a very wide range of sea conditions.

1.1.2 Findings

Initially we investigate the operation of the lamprey neural segment as a neural oscillator and we show that the performance of this oscillator is determined by the synaptic weights of the network. We also show that we can effectively connect the neural and mechanical models together, resulting in the neural output becoming phase locked to the WEC oscillation. Furthermore, we show that we can also optimise the neural weights and develop a neural network that is able to apply a latching[3] strategy to the single buoy WEC, producing near-optimal response in a range of wave periods. We also show that this evolved single buoy controller is able to produce good output in realistic waves, without requiring any prediction of the wave - a feat that is currently particularly difficult.

The mechanical model is now expanded in order to begin to approximate an articulated device. It is shown that in the configuration of two sections and one joint (the Cockerell raft) performance similar to the single buoy results is seen with good performance also shown in realistic waves. Further expanding the system to three sections and two joints interestingly showed that effective latching control is not possible in this configuration. It was seen that the interconnected lamprey CPG segments were unable to apply latching control to an articulated WEC primarily due to the discrete nature of the control method used, however this opens up the future possibility of using this neural technique to apply a more continuous method of control.

[3]Latching is a method of discrete control that involves locking the WEC displacement when the device reaches zero velocity and releasing a short time later. A full explanation is given in section 2.4.2

1.2 Original Contribution

This project aimed to investigate the hypothesis that:

Effective control implementation for Wave Energy Devices can be developed from (neuro)biological exemplars.

The choice of the lamprey as the (neuro)biological exemplar and of the Pelamis WEC as the artificial system are made for the reasons described above. The "fit" between the two systems appears to be good and encourages optimism that the hypothesis will be supported in this specific and important case study.

As with any body of research, this thesis builds upon a base of knowledge already in place. This work concentrates in two dissimilar areas, artificial neurophysiology and marine energy conversion and provides a contribution to knowledge through their combination.

This work is based upon a well proven and very simple approximation to a wave energy converter, a heaving buoy that incorporates a power take-off element in the tether. A technique known as latching is applied to this system which was first developed in 1978 by Budal and Falnes [1] in order to improve the efficiency of the power take-off mechanism. Any ideal control mechanism requires that the future wave elevation is known and this is clearly impossible with any control technique as it would require conditions to be fully constrained and predictable. Current near-optimal practical methods can, however, derive future wave elevation in real seas through prediction or measurement ahead of the device. This thesis investigates the possibility that by using a biologically inspired neural network, we may be able to remove some of the requirement for measurement or prediction when using a latching strategy in an irregular sea.

A repeat of the basic latching principles is presented in Ringwood and Butler's 2004 paper[7]. This also shows a very simple application of genetic algorithms to determine optimal latching parameters. Whilst introducing the principle of GA's to wave energy, this work only achieves values to determine latching parameters in a known regular sea. In contrast, this thesis shows the use of GA's to optimise a (neural) control system so that the device may apply latching in an unknown sea, thus operating without the need for wave prediction.

The principle of applying latching control in irregular waves without prediction is known as a "causal" strategy and a method of implementing this was presented by Korde in a 1999

paper [8] which presents a mathematical method of extracting the future device velocity from a combination of past and current movements with direct reference to latching. This thesis in essence provides a similar result to Korde but uses a significantly different method. It was found that there is a minimal amount of published work in this area, possibly due to the sub-optimal nature of such a resulting strategy.

The neurophysiological aspect of this work is solidly based upon models of the lamprey developed by Ekeberg and further expanded by Ijspeert. In addition, this work also makes use of similar evolutionary techniques that Ijspeert used to develop the synaptic weights of the lamprey CPG, although in this research the power absorbed by the device is used as the fitness, rather than the forward swimming speed.

The novel contribution of this thesis lies in the unique and very promising application of neuronal networks to apply control to a simple WEC device. It can be seen that there is very little precedent for this work, so it is difficult to identify exactly where this research sits. It can certainly be seen as a further expansion and abstraction of the work done by Ijspeert and Or by developing further the artificial lamprey model and illustrating an alternative application. It also sits firmly within wave energy as another expansion to the study of latching control by illustrating an alternative non-mathematical method to Hoskins' and more recently Babarit's work on determining optimal latching response.

Furthermore, this work presents the principle of a non-optimal control strategy that will work without requiring future knowledge of the wave height which offers an alternative and simpler method to that presented by Korde. Also introduced here is an attempt to produce latching in an articulated device, a principle which is mentioned by Babarit but not expanded. The results here suggest that this neuronal model would be far better suited to control of devices with a single control element.

In effect this thesis creates new ground by suggesting a novel application of biologically-based neural networks, creating a small branch in the area of wave prediction and control application. This will hopefully pave the way for further work in this area, including looking at the use of different control techniques, testing in more realistic conditions with more realistic devices and the further development of the neural parameters themselves to improve the overall performance.

1.3 Summary of Thesis

Chapter 2

Wave Energy Background

This chapter introduces the concept of marine energy and illustrates the great potential power that may be available in ocean waves. It explains some of the preliminary issues and principles that need to be addressed in the conversion of ocean waves and describes techniques that can be used to optimise the power transfer. The proposed method of control is outlined along with a full description of the mechanical model that will be used throughout this thesis.

Chapter 3

Neural Background

This chapter describes the biology of the lamprey and introduces the concept of centrally controlled locomotion. The neurophysiology of the central pattern generator is studied and its unique features are detailed. Following this, the text looks at how the lamprey CPG has been computationally modelled and simulated in the past and it describes how these models can be developed to potentially produce a controller for a WEC.

Chapter 4

Tools and Methods

In this chapter the basic experimental method and techniques are introduced. The numerical model for the neural network is introduced and a description is given of how we may combine this with the mechanical WEC model described earlier. Genetic Algorithms are introduced in some detail, illustrating some of the more advanced techniques that will need to be employed to evolve solutions, including outlining a multi-processing method to speed up execution.

Chapter 5

Preliminary Results

This chapter introduces the first experiments and shows how the neural and mechanical systems interact. It describes latching control for a single buoy initially and then illustrates how this can be developed into a wide-bandwidth controller that is capable of dealing with changing period

waves. Different wave periods and controller bandwidths are shown and the effect of varying the neuron time-constant and optimal damping value is investigated.

Chapter 6

Further Testing and Expansion to Multiple Degrees of Freedom

This chapter continues to test the wide bandwidth controller developed in the previous chapter, showing its performance in more complex and realistic waves and comparing it to uncontrolled optimal damping. The mechanical model is expanded into two units and its performance shown in a series of regular & irregular waves. The mechanical model is then further expanded into three buoys and the performance with these simple articulated arrangements is discussed with emphasis upon suggesting a number of ways in which to evolve successful controllers.

Chapter 7

Discussion and Conclusions

This chapter discusses the significant results and issues raised within this body of work and presents the the conclusions that may be formed from them.

Appendices

Appendix A

A full description of the operation of the simulator is provided in this appendix, including operating instructions and examples of configuration files required for operation.

Appendix B

This appendix includes the evolutionary algorithms featured in this thesis and describes the genetic operators used. The method, operation and use of the multi-processing environment is described along with a listing of the perl scripts and sample control files.

Chapter 2
Wave Energy Background

This chapter provides an introduction to the basic concepts behind ocean waves and introduces the principle of energy capture from these waves. Starting with a historical overview of the motivation behind the study of wave energy, this chapter looks at some of the important concepts required to understand capture of energy from ocean waves. Following this explanation, we look at the possible ways in which a device may capture energy and detail previous work done in this area. The concept of control is introduced next with the principle of latching, or phase control, being addressed in depth. Finally the modelling of a simple system in relation to this thesis is presented, along with a possible control implementation.

This chapter covers a great deal of work in the field of wave energy but does not provide an in-depth coverage of all the associated areas; what it does provide, however, is a coverage of the techniques and principles directly relevant to the ideas presented later in this thesis. An excellent overview of the field of wave energy can be found in chapter 8 of "Renewable Energy", by Les Duckers [9] or alternatively, a thorough treatment of the principles of wave energy extraction can be found in Johannes Falnes book, "Ocean Waves and Oscillating Systems"[10–12]. These publications provided a significant amount of background for this chapter.

Whilst a complete coverage of the whole subject of wave energy is not feasible, references will be provided to guide the reader to the most significant works in the appropriate areas.

2.1 Background

Conversion of ocean waves into usable energy has been of practical interest for over one hundred years. Sea waves provide a very tangible force that has been observed since pre-history, however the practicalities of capturing this energy have been far from straightforward. Modern interest in ocean wave energy conversion started around 1973 with initial work being carried out at Edinburgh and Trondheim Universities, amongst others. This interest in wave energy as

a renewable resource was triggered largely through the energy crisis of the 1970's, with many people being particularly concerned with possible replacements for fossil fuels. Much research took place during this initial period, and the basis of modern theory was laid down with many lessons learnt and many failures observed. Research funding in the UK was cut in March 1982 even though the research at that time was exceptionally promising[13], the decision being made to rely on nuclear power to provide clean cheap energy. It has been seen since that this decision was critically flawed when the real cost of nuclear power became evident. Since the late 90's, however, interest has resumed in wave energy mainly due to two factors:

- An unhealthy dependence upon a dwindling supply of fossil fuels
- The Kyoto protocol and resulting CO_2 emissions targets

These factors resulted in demand for low carbon, renewable sources of electricity and in the marine energy sector alone, have triggered some very promising technologies and devices. Many of these are now at the full scale prototype level and although none have yet been deployed on a commercial scale, this is certainly imminent.

2.1.1 The Wave Resource

The surface of the earth consists of $\frac{2}{3}$ water and the energy contained in the movement of this medium is huge. This is divided into movement through gravitational interactions, (tides) and through waves resulting from winds blowing across the surface. Both wave and tidal energy can be captured and can in theory generate enough power to have a significant contribution to global energy production. This thesis considers only the extraction of energy from waves and does not consider tidal extraction which would require a very different method for capture.

The energy contained in this resource around the globe can be seen in fig 2.1. Concentrations of waves with annual average power densities exceeding 80kW per meter of wave crest can be observed in northern latitudes. It is important, however, to realise that the nature of the wave climate means that the figures in figure 2.1 are year-round average figures; certainly in these locations extreme conditions will be encountered that may be many orders of magnitude greater than the figures given here. Indeed the seasonal variation will tend to much larger waves in the winter months, a particularly convenient effect in northern latitudes as the energy demand will

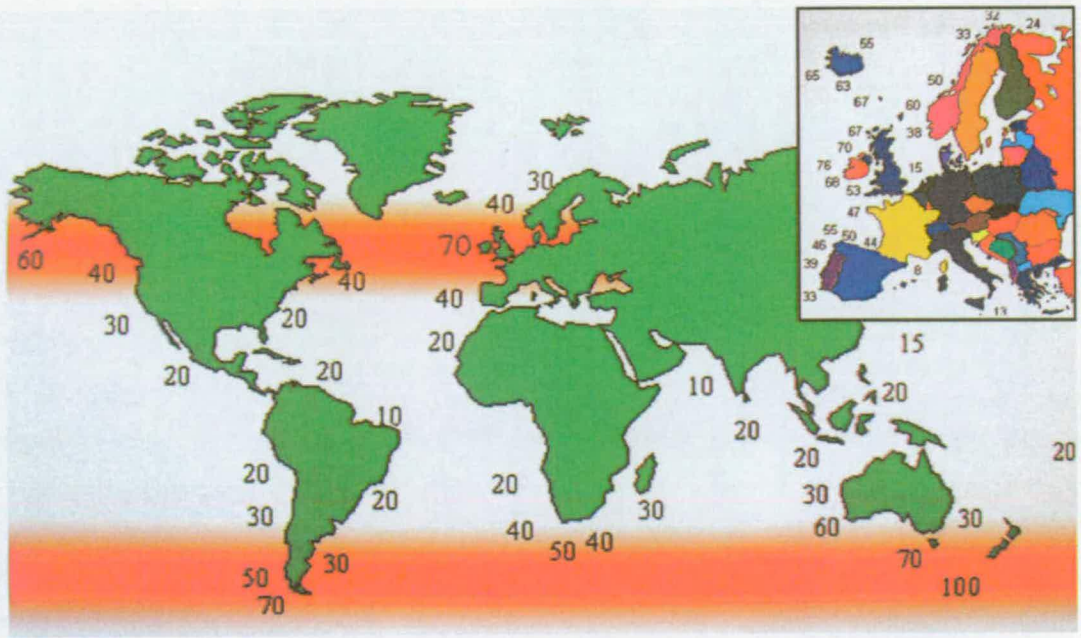


Figure 2.1: *World wave energy resource. Figures show annual average available power in kW per meter of wave crest.*

also be higher at this time of year[1]. The year round variation for a site off the Hebrides in northern Scotland is shown in figure 2.2.

2.1.2 Waves as Condensed Energy

Ocean waves consist of a combination of many different types, of which all are directly or indirectly formed through winds blowing across the sea surface. These winds are created, directly and indirectly, through the solar heating of the atmosphere. It can be seen that as the flow of energy propagates from solar to wind to wave the density of this energy is compressed (see table 2.1).

As energy is transported between media it results in an increase in density of around 30 times. This results in particularly high energy concentration which can be extremely beneficial in order to enable economical capture. However, this increase in density also results in a power spectrum

[1]This refers to the location mentioned in figure 2.2 and applies to areas in which the requirement for heating and lighting during the winter months results in a significant increase in electricity demand relative to the summer months.

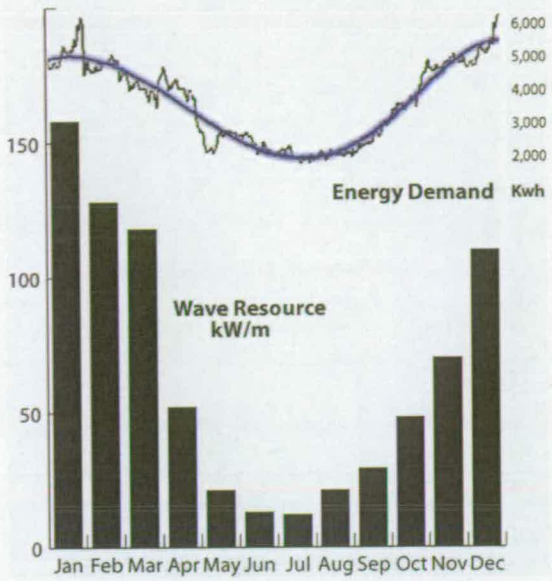


Figure 2.2: *Annual seasonal wave variation for an area just off the Hebrides [12]. Powers shown are average kW per meter of wave crest. The upper line illustrates the annual energy demand in kW for the Scottish town Lochgilhead[14]*

Resource	Average Intensity
Solar	100-200Wm^{-2}
Wind	400-600Wm^{-2}
Wave	2 - 3 kWm^{-2}

Table 2.1: *Resource Energy Density[12]*

where peaks are many magnitudes higher than the average (see figure 2.4). The nature of the resource means that any cost-effective energy capture device must be engineered to survive the very high energy peaks whilst still maintaining a high efficiency.

2.2 Waves on Water

In order to describe the nature of waves, it is best to think of them as a movement of energy across the surface of the water, as the water particles themselves to not propagate along with the wave, rather they travel in orbits as the wave passes. Considering these orbits we can derive the equations of motion that govern the major characteristics of waves.

It is important to know the energy content of these waves and just how the energy is distributed

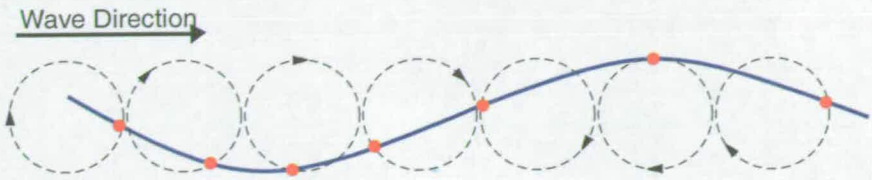


Figure 2.3: *Circular particle movement in a regular wave. Particles are represented by red dots and their orbits are represented by the circles.*

over the wave spectrum. From this we can see the different behaviours of waves on water and how they may interact with objects designed to harness this energy.

If we consider that the device will be used only in *deep water*, which is classified as a depth greater than $\frac{\lambda}{3} \rightarrow \frac{\lambda}{2}$[10], then we can state the velocity of propagation of a wave $Acos(\omega t - kx)$. Assuming that $\omega = 2\pi f$ and knowing that the angular repetancy (wavenumber-k) $= \frac{2\pi}{\lambda}$, then using $c = f\lambda$ this gives:

$$c = \frac{\omega}{k} = \frac{g}{\omega}^{\dagger} \tag{2.1}$$

However, water waves are dispersive. That is, waves of different frequencies and wavelengths will travel at different speeds. This dispersion relationship is given by $\omega^2 = gk$. Therefore the speed, or as it is more correctly known, the phase velocity v_p is given as:

$$v_p = \sqrt{\frac{g}{k}} \tag{2.2}$$

$$\lambda = v_p T = \frac{g}{2\pi} T^2 \tag{2.3}$$

However, due to this dispersion relationship, if waves of different periods are superimposed, then it can be seen that the resulting group of waves will propagate with a different velocity. This can be shown by using two waves of different frequencies ω_1 and ω_2, and letting $\omega_1 = \omega - \Delta\omega$ and $\omega_2 = \omega + \Delta\omega$. Therefore the sum of these waves can be written as:

$$\eta = 2a\ cos(\Delta\omega t - \Delta k x) cos(\omega t - kx) \tag{2.4}$$

If we consider the term derived from the summation of these frequencies: $2a\ cos(\Delta\omega t - \Delta k x)$ This is the energy carrying component of the resulting wave and will propagate with the group

$^{\dagger} c = \frac{g}{\omega}$ is derived from the dispersion relationship for waves on deep water; $\omega^2 = gk$ [10]

velocity (vg):

$$v_g = \frac{\Delta\omega}{\Delta k} = \frac{d\omega}{dk} = \frac{g}{2\omega} = \frac{1}{2}v_p \tag{2.5}$$

Therefore it can be seen that the group velocity is half of the phase velocity. This is a significant result, as it shows that the propagation of the group and hence the energy content of the wave is less than the speed of an observed individual wave.

2.2.1 Real Ocean Waves

As is obvious though observation, real ocean waves comprise a complex interplay of regular waves with different period, length and direction. These waves superpose to produce a spectrum that, in general, features a dominant period and wavelength. Although real ocean waves are stochastic and very difficult to predict accurately, they can be modelled with suitable spectra generated from a random seed. In 1964 Pierson and Moskowitz[15] developed a spectrum that would model a fully developed, wind-driven sea. This enables us to produce an estimate of a real sea and facilitate the design of structures, ships or more appropriately, wave energy converters.

$$S_{(\omega)} = \alpha g^2 \omega^{-5} e^{-0.74(\omega_o/\omega)^4} \tag{2.6}$$

Shown in equation 2.6 the Pierson-Moskowitz spectrum is still useful today although it has somewhat been superseded by the JONSWAP spectrum. In 1973, through the Joint North Sea Wave Observation Project, Hasselmann et al. [16] found that the wave spectrum is never fully developed. It continues to develop through non-linear, wave-wave interactions even for very long times and distances. Therefore a spectrum of the form shown in equation 2.7 was proposed.

$$S_{(\omega)} = \alpha g^2 \omega^{-5} e^{-1.24(\omega_o/\omega)^4} \gamma^r \tag{2.7}$$

$$r = e^{-\frac{(\omega-\omega_p)^2}{2\sigma^2\omega_p^2}} \tag{2.8}$$

The JONSWAP spectrum is similar to the Pierson-Moskowitz spectrum except that waves

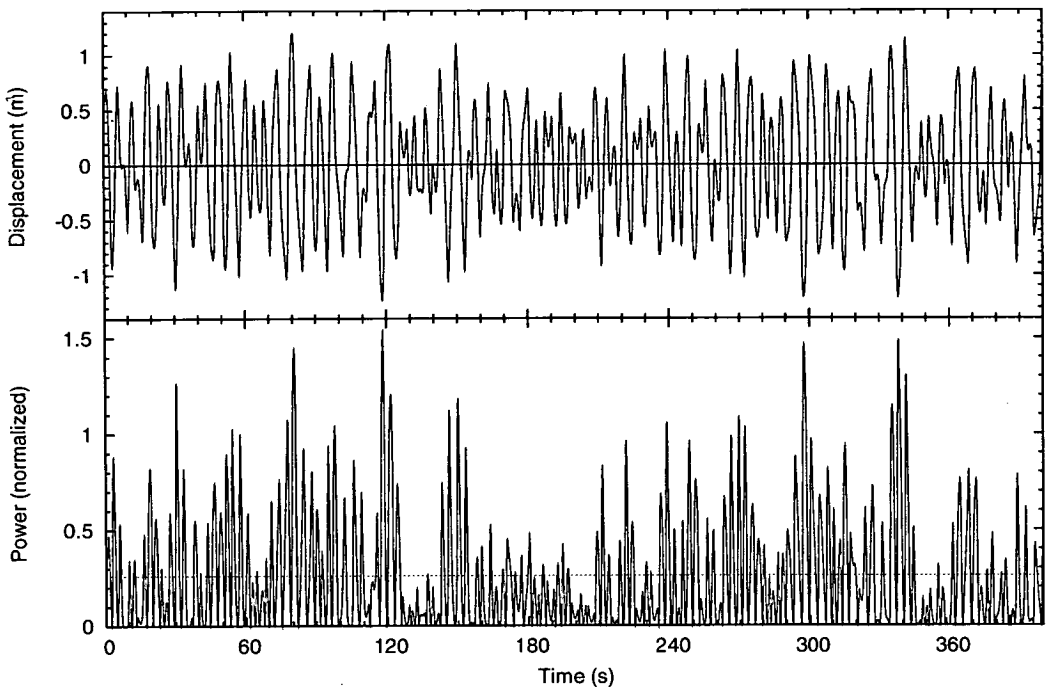


Figure 2.4: *Wave height against time for a wave derived from the Pierson-Moskowitch spectrum in equation 2.6. The lower plot shows the relative instantaneous power for this wave. The horizontal dotted line indicates average power.*

continue to grow with distance (or time) as specified by the α term, and the peak in the spectrum is more pronounced, as specified by the γ term.

A time representation of a Pierson-Moskowitz spectrum is shown in figure 2.4, this figure also shows the instantaneous power content.

2.2.2 Energy Content

It can be seen [10] that, within one wavelength, the kinetic energy contained within the wave must equal the potential energy. Therefore $E_t = E_p + E_k = 2E_p$ and can be written as:

$$E_t = 2E_p = 2\,\frac{\rho g}{\lambda}\int_0^{\lambda/2} \eta\, dx = \frac{1}{2}\rho g \left|\eta\right|^2_{max} \tag{2.9}$$

Power contained within the wave is defined as the flow of energy per unit length of wave front, and as the energy moves with the group velocity this gives $J = v_g E_t$. If we also substitute eqn 2.3 into eqn 2.5 and use the RMS value of wave elevation H_{RMS}, after manipulation eqn 2.9 becomes:

$$J_{(Wm^{-1})} = \frac{\rho_w g^2}{4\pi} H_{RMS}^2 T \tag{2.10}$$

This provides the maximum power per unit of wave crest for a given period and height. It is important to note that the power increases with the square of the wave height and this relation is illustrated in figure 2.4 with the power content of the PM spectrum shown in the lower plot. It is significant to note the difference between the average and the peak power as this is a key characteristic of wave energy and any WEC must be designed to take this into account. Salter [17, 18] discusses the importance of power limiting and storage in WEC's in order to improve performance in such an environment.

2.3 Capturing the Power

We now will consider what happens when the waves interact with a floating body. If we make the dimensions of the body small with respect to the wavelength they can be neglected, save for the obvious effects of the volume, buoyancy and hydrodynamic forces. This in effect means we can neglect more complex effects based on the physical dimensions of the body. This assumption is commonly known as the "Small Body Approximation".

The floating body now has a buoyancy dictated by its volume, and defined by Archimedes Principle (eqn 2.11). The force in the vertical direction, is opposed by the mass of the body. This balance of forces creates a spring effect, which can be observed by the resulting oscillation if the body is displaced by a small amount and then released. This oscillation will then be damped due to hydrostatic/hydrodynamic forces. The resulting system obeys the simple harmonic motion of a mass-spring-damper (eqn 2.12).

$$Buoyancy(F_b) = \text{submerged volume} \times \rho_w \times g \tag{2.11}$$

$$mass \times \ddot{z} = -damping \times \dot{z} - k \times z \tag{2.12}$$

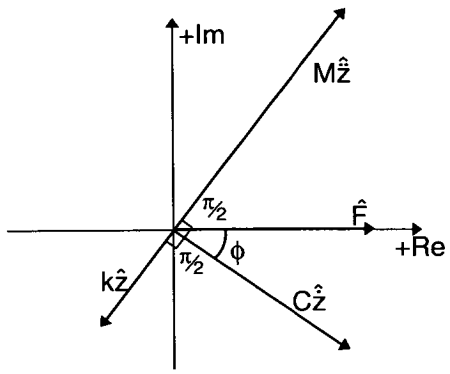


Figure 2.5: *Phasor diagram showing the $\frac{\pi}{2}$ shift between components. The over hat represents complex notation.*

Here we assume z is displacement in the vertical direction, k is the spring constant or buoyancy, g is the acceleration due to gravity and ρ_w is the density of water. The over-dot is used to denote differentiation with respect to time. The relationship between these components can be illustrated in complex form in figure 2.7.

The hydrodynamic forces acting upon the body are commonly defined as added mass and added damping. These forces are the result of a body moving in water. The added mass is a result of wave diffraction and can be visualised as the mass of water a moving body 'drags' along with it. The added damping, more accurately known as radiation resistance can be described as the force resulting from the oscillations emanating *from* the body.

Both the added mass and damping are related to the physical shape of the body and are frequency dependent, however these are first order effects and cannot be neglected by the small body approximation. For the purposes of modelling the hydrodynamic co-efficients of this simple geometry have been calculated using the WAMIT [19] program and are shown in figure 2.11.

2.3.1 Principle of Energy Extraction

Initially shown by Evans in 1976 [20], in order to extract the energy from an ocean wave we must understand the paradox that states: *"In order to destroy a wave, we must create a wave"* [10]. By destroying a wave we are capturing its energy and in order to do this we should create a wave that will destructively interfere with the incident wave. Therefore a good absorber

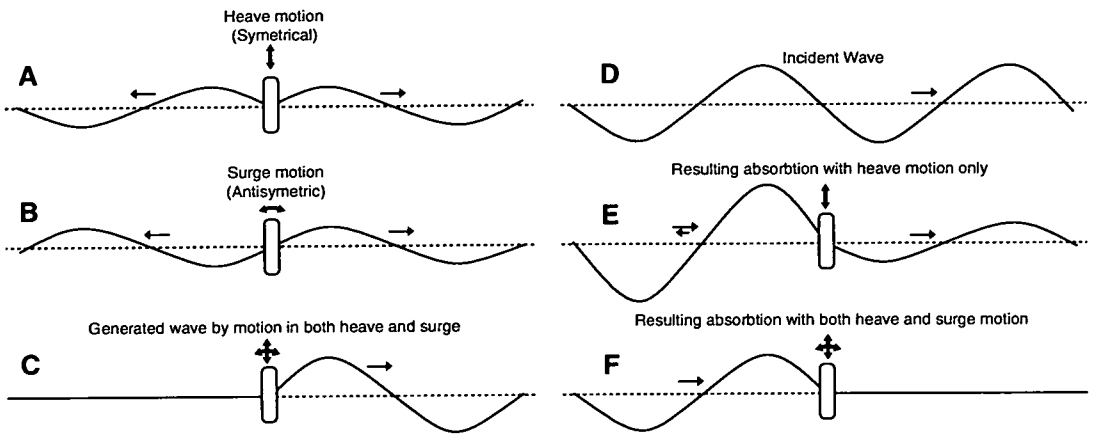


Figure 2.6: *The process of absorbing a wave.[10]*

of waves is also a good creator or waves. Figure 2.6 illustrates this concept.

A & *B* shows the waves generated when a body is restricted to move only in heave (A) or surge (B). It can be seen that if we combine these motions, the resulting superposition will give us *C*. This shows that the combination of the symmetrical and asymmetrical motion has cancelled out the radiation propagating to the left, leaving only a wave propagating to the right.

If we now consider the incident wave shown in *D*, we can see that if we combine this with the wave generated by the heave motion, only 50% of the incident wave will be absorbed by the body *E*. This would also be the same for any body only operating in one dimension. However if we now combine the wave generated through the combination of heave and surge motion (*C*) with the incident wave, we can see that 100% of the incident wave can be absorbed(*F*).

By carefully controlling the motions of a floating body it can be seen that it is possible to absorb all of the incident wave energy. Although the illustration in fig 2.6 considers a case in only in two dimensions, in reality there are many more factors to consider.

2.3.2 Previous Work

Much work has been done on device development since the mid seventies when wave energy first emerged as a quasi-viable energy source. Since Salter's landmark 1974 paper [21], the theoretical fundamentals of wave absorption soon appeared independently in publications by Evans [20], Mei [22] and Newman [23]. These theories showed that a body could absorb

more energy than was incident upon its width. Newman [23] showed that in an optimal simple case, a device is capable of converting a maximum of $\lambda/2\pi$ of the incident wavefront. This is a significant result as it eliminates the device dimensions from the equation. Evans [24], and more recently Pizer [25](with specific reference to point absorbers) illustrated that as the excursion of the body must adjust to accommodate this power, practical excursion constraints should be considered.

Different types of WEC can traditionally be placed into one of three classifications: point absorber, terminator and attenuator. The point absorber can be the simplest of all the designs and commonly features small dimensions relative to the incident wavelength, taking advantage of the small body approximation. In general, due to their simplicity, the point absorber is the simplest device on which to implement control (see section 2.4.2).

The terminator design, as the name suggests, has the principal axis parallel to the wave front and attempts to absorb all of the incident energy. Devices in this category include Salter's duck [21] and more recently the wave-dragon overtopping device [26].

The attenuator design commonly has the axis perpendicular to the wave front and can be seen to be inherently power-limiting. The most prominent design in the category is Ocean Power Delivery's PELAMIS [5, 27, 28] which consists of a series of four articulated cylinders, interconnected by hydraulics, positioned perpendicular to the wave front. Although a new device, the original concept was proposed by Cockerel (as reported in [29]) which consisted of trains of interconnected rafts whose relative angular movements generate power. This concept was modelled by Newman [30] and Haren & Mei [31] who studied the efficiency and optimisation of such a design.

Many devices do not fit neatly into these classifications, such as the Bristol Cylinder (as described by Salter [17]) which consists of a large floating cylinder held under the water and parallel to the wavefront so that it moves with the circular orbits generated by waves, as shown in figure 2.3.

2.4 Control

The principles of energy extraction illustrate a number of complexities that must be addressed in order to attain optimal efficiency. Mainly these inefficiencies arise as the extraction system

is frequency dependent, thus having a different response to different waves. Thus, in order to tune such a system in a given wave environment, various parameters of the device need to be adjusted. The simplest method is to tune the device to the most common wave environment at the proposed site and accept that for some of the time the device will not be operating within the optimal region. Alternatively it may be possible to design a device that will allow certain parameters, such as power take-off damping, inertia etc to be adjusted - enabling the device to adapt to changing sea conditions. This is a common technique as it enables the device to operate at the maximum of its efficiency for more of the time, increasing its plant capacity factor[2]. In both of these methods, the device parameters are static for at least a number of wavelengths, usually much more, and can be classed as *tuning*[3] of the device rather than *control.* Control on the other hand describes the tuning of the device on a cycle by cycle basis and can result in a significant increase in the rated efficiency of the device. Generally, applying control is a very effective way of increasing the efficiency of a device without significantly contributing to its cost. Therefore much work (see section 2.4.2) has been done in this field. In order to understand the most prominent methods of control we must first introduce the basic principle of resonance.

2.4.1 Principle of Resonance

A very well known principle, resonance describes the exchange of energy between two systems at a specific frequency. To expand, we can first define the natural period of oscillation of simple harmonic motion, such as seen in the system here, (ω_0) which can be found using $\omega_0 = \sqrt{\frac{k}{m}}$. This shows the dependence of the natural frequency upon mass and buoyancy. Now consider the body being exposed to an external wave force; when the natural period of the excitation force coincides with the natural period of the body, ω_0, maximum energy will be transferred between these two devices. At this point the energy will be dissipated by creating very high excursions on the body. If we add additional damping into the system, for example with a tethered damper to the sea bed, we can absorb the energy imposed upon this body and restrict its excursion. This damper can be considered the power take-off device and will be referred to

[2]The plant capacity factor (PCF) is given by the ratio of the integral of energy produced to the device rated power. For a particular resource it describes the amount of time the device is operating at its rated (maximum) capacity and is given in percent. Therefore a 1MW rated device device with a 50% annual capacity factor will produce a continuous 500kW *average* power over a year.

[3]A 'tuned' device, as mentioned here is strictly classed as an *uncontrolled* device and will be referred to as such in this thesis

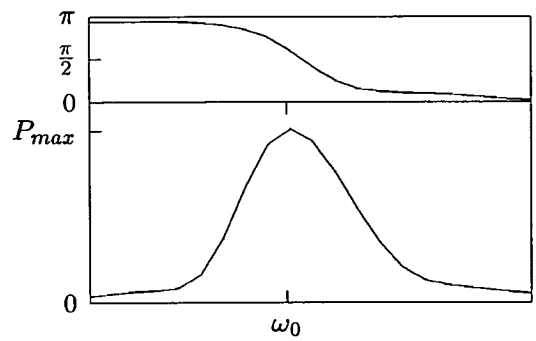


Figure 2.7: *Power and phase response against wave period for an uncontrolled system. The phase is measured between the device and wave displacement. Above and below ω_0 the fall off of absorbed power is in line with the phase shift approaching either π(when $\omega > \omega_0$) or zero ($\omega < \omega_0$)*

as C_{PTO} in this thesis.

Whilst it is clear that at resonance optimal energy transfer between a device and the wave will take place, above or below this frequency significantly sub-optimal transfer will occur. Due to the non-regular and unpredictable nature of real ocean waves (see fig. 2.4) this means that for the majority of time a simple device will be operating at a significantly reduced capacity.

Budal & Falnes stated in 1978 [1] that this point of (resonant) maximum power transfer was characterised by a phase difference of $\frac{\pi}{2}$ between the wave and the device displacement. Figure 2.7 illustrates this relationship between phase and power. They suggested that if this phase shift could be achieved at frequencies away from resonance it would result in maximum possible power transfer.

2.4.2 Control

As reported previously control can be described as modifying the device parameters on a cycle by cycle basis in order to increase the conversion efficiency. The method in which this is done depends upon the control technique. Various research has been done on different control methods and their suitability to various seas, Korde [32] in his review article provides a general overview of the different control techniques, however a more in-depth review can be found in Falnes [33].

In theory, many of the proposed control methods [1, 21, 34] can produce near optimal power in any sea, however they require advance knowledge of the incoming wave. This results in what can be termed a 'non-causal' system that requires future knowledge of the system in order to achieve maximum performance [35, 36]. Although this is a significant problem in control system design, wave-prediction, measurement and forecasting can be used in order to combat this.

The two most significant *control* methods developed are *Reactive* and *Latching* control and are discussed below:

Reactive Control

Reactive, or continuous control was developed at Edinburgh University through early work on the duck [21] and involves modification of device parameters such as inertia, spring, damper etc. It is a continuous phase control method and involves the continuous application of forces to the device, an excellent overview is provided by Salter [18]. The applied forces to the device are in phase with both inertia and spring as inertia resists acceleration and spring resists displacement. This control method adds to or cancels the spring and inertia terms, ensuring that the device performs as a pure damper, absorbing the available energy, rather than re-radiating it. This action sometimes requires that energy is put into the system, however it is normally repaid highly with the increased efficiency.

An extension of reactive control is complex conjugate control, this is a more general implementation of control derived mathematically. This method seeks to cancel all undesirable spring and inertia at all frequencies and in theory, by using prediction of the incoming wave, an almost 100% flat efficiency curve can be produced [34].

Latching Control

Latching, or more correctly, discrete phase control, was developed in 1978 by Budal and Falnes. Budal [1] proposed that optimal phase may be achieved through the fixing and releasing of the device at discrete points through the wave cycle. This method is termed "latching" and was also proposed independently by Jones [37] and French [2]. The method involves the device being held at its extremities of displacement for a short time so as to induce the correct phase difference between the device oscillation and the incident wave. Figure 2.8 shows how the

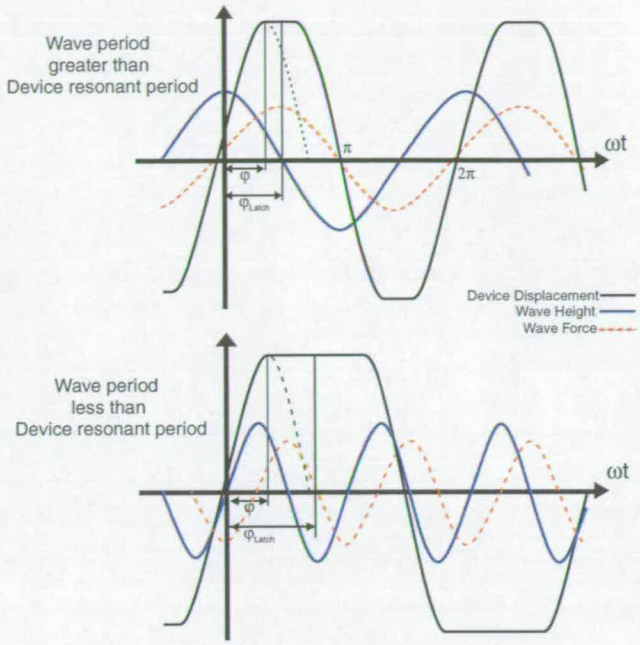


Figure 2.8: *Latching strategy. Upper diagram shown the common configuration where the natural period of the device is lower than the incident wave. The lower diagram shows how latching may be implemented when the wave period is shorter than the incident wave.*

displacement of the device is "held" at the maximum positive displacement until the force upon it is maximum. At this point it is released and moves until it it reaches maximum negative displacement, whereby it it is again fixed. This results in the optimal phase shift of $\frac{\pi}{2}$ being achieved between the device and the wave. However as the control strategy implements phase control by *increasing* the phase shift, it can only be used effectively in this configuration for frequencies less than resonance (see fig 2.7). More recently, Babarit et al [4] showed that latching could also be used below resonance, this results in a latching period of greater than one half period and therefore only one movement, or less, is allowed per cycle (see fig 2.8).

The discrete nature of latching control means it is difficult to mathematically determine an optimal strategy. However Hoskin and Nichols [3] developed an iterative computational method to calculate the optimal performance of this method, then later [38] showed a mathematical method for solving this problem using Pontrygrains Maximum Principle.

As with any control method, latching requires knowledge of the incoming wave in order to calculate the correct release points [35], hence any optimal solution must require future

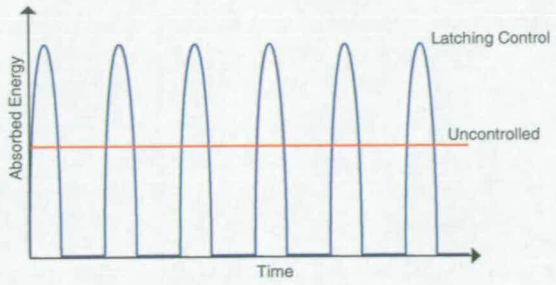


Figure 2.9: *Energy absorbed against time for a latching strategy, compared against an uncontrolled response.*

knowledge and is known as 'non-causal'. This is a particular problem when dealing with irregular spectrum waves as these must be predicted or detected in advance in order to ascertain the correct latching points. Conversely, a system that can calculate the correct latch and release points *without* requiring future wave data can be termed 'causal'. Korde [8] shows that causal control may be possible by extrapolating from past and present data and more recently it has been shown that it is possible to implement a causal strategy [39] that although non-optimal, can still provide a significant increase over an uncontrolled system.

The differences between continuous and latching control can be highlighted if we look at the power take-off (C_{PTO}) from these two systems. Figure 2.9 shows the cumulative developed power over time and clearly highlights the discrete nature of latching.

2.5 Numerical Modelling

In order to be able to study the effects of interactions between a device and water waves, we must create a model that will react appropriately. The purpose of this thesis is not to create a full numerical model of a wave energy converter, but to study a novel method of control. Thus work will concentrate on a very simple point absorbing device that will capture the richness of a real device while enabling fast simulation for demonstration of control methods.

2.5.1 Single Buoy

The structure of the simple point-absorber model is shown in fig. 2.10. It consists of a long cylindrical buoy tethered to the sea floor, with the power out-take being incorporated into

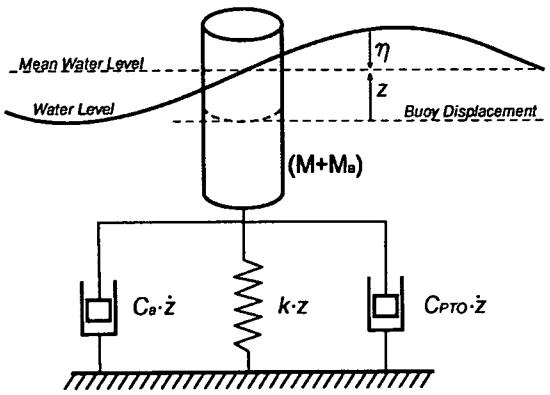


Figure 2.10: *Single buoy model.*

this tether. The height of the buoy is chosen so as not to become completely immersed or revealed and so maintaining a linear force profile. Each of the significant terms are shown on the diagram. The following parameters are selected to give the model a natural frequency ω_o higher than the majority of waves we will be working with, this is due to the application of a latching strategy at a later stage.

Diameter of Buoy = 1.65 metres

Height = 10 metres

Mass = 10.6 tonnes

50% submerged at equilibrium.

The force on the buoy when submerged, given by Archimedes principle, is equal to the weight of the fluid displaced. This can be shown as:

$$Buoyancy\ Force = \pi r^2 \times \rho_w g \times h = h(\pi r^2 \rho_w g) \tag{2.13}$$

Here ρ_w is the density of water, g is the acceleration due to gravity, r is the radius of the buoy and h is the height of the buoy.

If it is assumed that the waves will be of a small enough magnitude so that the buoy will never be fully submerged or removed from the water, we can assume that the restoring force on the cylinder is linear. With this assumption we can therefore neglect the height of the buoy and thus we assume that at equilibrium the buoy is 50% submerged. In an oscillating system such

as this the restoring action is primarily from the spring term, $-kz$. Therefore:

$$-k\dot{z}_{(t)} = -z_{(t)}(\pi r^2 \rho_w g) \tag{2.14}$$

We must also consider the excitation due to incident waves, if we determine the water height from equilibrium to be η then the excitation force on the cylinder will be:

$$Fe_{(t)} = \eta_t(\pi r^2 \rho_w g) \tag{2.15}$$

It can be seen that when the buoy is at equilibrium the excitation force equals the restoring force; therefore as $(\pi r^2 \rho_w g)$ is a constant, $\eta = -z$, hence this shows the buoy is resting at the surface.

2.5.2 Hydrodynamic Effects

In order to correctly model the motion of the device, we must take into account the major hydrodynamic forces that will act upon the buoy.

To determine the correct values for the added mass, M_a and the radiation damping, C_a, a simulation of the buoy geometry was carried out using WAMIT software (fig. 2.11). The resonant frequency (ω_0) of the system is shown and it can be seen that the radiation damping decreases to a minimum at this point, illustrating minimum losses. It can be observed that this occurs at around 3 seconds, which is in agreement with the calculated value of $\sqrt{\frac{k}{m}} = 2.99s$.

2.5.2.1 Added Mass

Added mass is an effect resulting from the waves interacting with the device, this physical effect is manifested by the tendency of any moving body in water to drag some volume of the fluid along with it. This effect contributes to the mass and inertia of the system and is frequency dependent. A plot of this for our system is shown as the upper trace in fig. 2.11.

2.5.2.2 Radiation Damping

Also termed added damping, this term models the effect of the waves radiated away from the buoy. It can be described as; 'the waves generated by operating the buoy in the absence of an

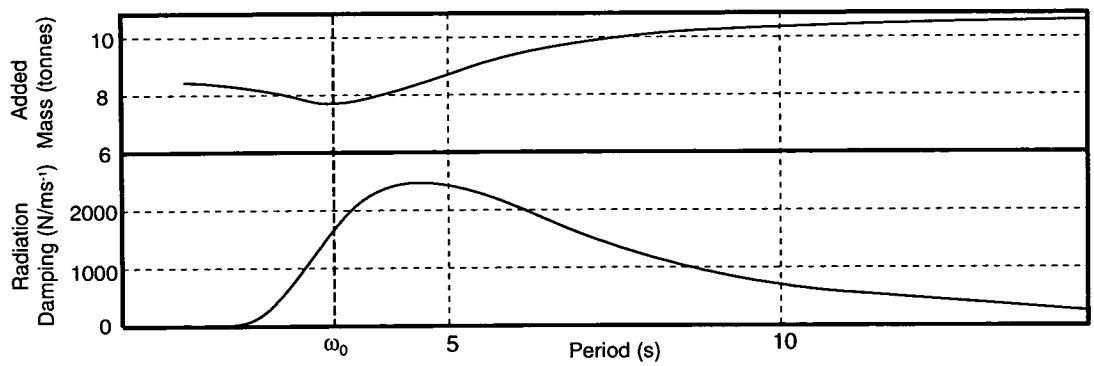


Figure 2.11: *Hydrodynamic parameters for a cylinder of radius 1.65m and height 5m, constrained to heave only.*

incident wave'. Although at first this may appear to be a loss term, this effect is in fact required for wave absorption. In fig 2.6 when the radiated wave (from a heaving body) shown in 'A' is combined with an incident wave shown in 'D' the resultant superposition is show in 'E' and clearly shows that this results in an reduction (absorption) of 50% of the transmitted incident wave. This term is frequency dependent and is shown as the upper plot in 2.11.

2.5.2.3 Hydrodynamic Approximation

It was necessary to develop a system that has no future knowledge of the wave and it is convenient that we approximate the frequency dependent quantities of added mass and added damping to non-frequency dependent static values. This is a significant assumption, however it was seen that this proved to be adequate for the requirements of this thesis, as the purpose of the model is to investigate the feasibility of a control method, rather than its refinements. As the majority of the investigation with this system takes place with wave periods much longer than resonance, (5-10s range) the error in approximating to static values is somewhat reduced (See fig 2.11). The resulting constants used in the model are shown in table 2.2.

Added Mass	8800kg
Added Damping	$2200 N/ms^{-1}$

Table 2.2: *Hydrodynamic Parameters.*

2.5.3 Developed Power

In order to extract power from the system it is necessary to apply a power take-off device. This is modelled simply as a damper and is denoted by C_{PTO}. Strictly speaking there should also be a loss term, which would also be modelled as a damper C_{loss}, however for simplicity, this has been set to zero. The instantaneous power developed in C_{PTO} can be calculated as $P_i = C_{PTO} \cdot \dot{z}^2$. This shows the power varying with the square of the device velocity.

2.5.4 Equation of Motion

As we have stated in eqn 2.12, the format for the overall equation of motion is straightforward. We can therefore incorporate the values and quantities discussed above. The mass of the system (M) is now augmented by the hydrodynamic effect of added mass (M_a) to give $M + M_A$
The damping within the system, previously referred to as C is now split into added or radiation damping (C_a) and the usable power take-off damping (C_{PTO}).
From eqns 2.14 & 2.15, the wave forces are decomposed into the hydrostatic spring effect, k and the force due to an incident wave, F_e.

This gives the resulting equation of motion as:

$$\ddot{z} = \frac{F_{e(t)} - C_a\dot{z} - C_{PTO}\dot{z} - kz}{(M + M_a)} \tag{2.16}$$

2.5.5 Control Implementation

This thesis will consider only latching control *(section 2.4.2)*, as this is the simplest method to interface with the neural network which will be considered later in section 3.

In order to apply effective latching control within this model, some method of fixing ("latching") the displacement at the extremities of its cycle was required. For simplicity this was implemented by switching the the value of C_{PTO} between its normal power-generating level and a much higher level that would resist any device movement. However, this method has the drawback that when C_{PTO} approaches ∞ the equations become stiff. This is due to the fact that the scale of the $C_{PTO} \cdot \dot{z}$ term becomes a considerably larger magnitude than the other terms in the equation. Although this is still a stable equation with a valid solution, this can lead to problems when solving computationally, such as requiring excessively small

time-steps or possible failure to find a solution. Although in this situation it is possible to use a different method to solve a 'stiff' equation, it was found, through trial and error, that using a value of $C_{PTO} = 10^7$ was an adequate compromise to hold the device in a 'latched' position without making the equations of motion stiff.

2.6 Chapter Conclusion

The awesome power present in ocean waves has been known for centuries, however the engineering challenges required to capture this energy have been significant. Recently, driven by the necessity to reduce global CO_2 emissions, advances in off-shore engineering have fostered the development of new devices that look set to make commercial wave energy a reality.

Development of control techniques is an effective way to increase the efficiency of wave energy devices, helping to reduce the cost per kWh and increase the capacity factor of the device. Of these methods continuous and discrete control are the most developed techniques, however the stochastic nature of the wave environment requires that any optimal control method have a future knowledge of the wave elevation. The impossibility of this requirement has lead work in developing prediction methods or 'causal' variants in the hope that this will aid the adoption of control in real devices and will help to improve the economic feasibility of the WEC.

This chapter has introduced latching and has highlighted the benefits of this type of discrete control. By locking and releasing at the extremities of movement a device's phase can be adjusted so that it is $\frac{\pi}{2}$ shifted from the exciting wave. It is shown that this condition is the *requirement* for optimal power transfer. It is worth considering that the implementation of latching control in a real device would not be without significant engineering hurdles. If we consider the forces required to lock, hold and release a full-scale buoy, which may have a mass of several hundred tonnes, we can see that they would be substantial. Although the engineering to implement such a system would admittedly be difficult, the performance gains as a result could offset this.

The simple model described here, although it is understood that it will not be strictly accurate, does provide an adequate approximation to enable study of latching control methods. This model in its single and articulated configuration has been discussed with engineers at Ocean Power Delivery (Developers of the Pelamis WEC.) and it has been agreed that it is possible to

capture enough of the salient features of a real or articulated WEC to be a useful and credible vehicle for a proof of concept study such as this. The simplifications made are necessary to create a system that is able to model the response of single or articulated WEC while requiring no prior knowledge of the sea state. This simplicity will enable fast simulation and will facilitate investigation into novel neural control strategies.

Chapter 3
Neural Background

This chapter introduces the inspiration behind the neural control method proposed later in this work. Starting with the biological method of animal locomotion, we look at how this is controlled, introducing the concept of the *Central Pattern Generator (CPG)*. The main subject of study, the lamprey, is then discussed, its locomotion is introduced and its particular unique characteristics are brought to light. Following this we introduce how biological networks may be modelled with particular attention to the lamprey CPG. Ekeberg & Ijspeert's' neuron model is then introduced and we discuss how this model can be modified and applied to the extraction of ocean wave energy.

3.1 Introduction

Biological neural networks describe a collection of electrochemical neurons that can commonly be found in the brain and central nervous system. These networks are also commonly referred to as *'neuronal networks'* in order to distinguish them from the computational *artificial* neural networks (ANN). The biological neurons are linked together with synapses that carry electrochemical signals from the axon of the pre-synaptic neuron to the dendrite of the post-synaptic neuron. The properties of neuronal networks have been extensively studied over many years in order to gain a better understanding of the structure of the brain and central nervous system.

In contrast, the artificial neural network bears little resemblance to its biological counterpart. Based in principle on the biological neuronal network, it is a example of a connectionist computational approach. It comprises an assembly of very simple (in contrast to the biological model) interconnected processing elements whose functionality is a result of the weights of these interconnections. Nevertheless this approach has a fantastic versatility and the ANN has become a feature in a large number of processing methods such as the multi-layer perceptron, Boltzmann machine, recurrent neural networks and feed-forward neural networks.

It will be seen that this thesis deals exclusively with biological neural networks although it does use the terms neural network and neuronal network interchangeably. Many of the methods presented here have been greatly influenced by the field of ANN, such as the connectionist models described later, however these ANN techniques are used as inspiration to simplify the investigation and modelling of neurophysiological behaviour, so can firmly be considered as biological neural networks.

3.1.1 Auke J Ijspeert's Thesis

This thesis uses Auke J Ijspeert's doctoral work [40] as the inspiration for the neural control method presented. In his thesis Ijspeert investigates and develops an artificial neural model of the lamprey, building upon on work done by Örjan Ekeberg [6]. His work implements Ekeberg's hand-crafted connectionist model of the lamprey spinal cord, using Sten Grillner's well known and well documented biological structure of the lamprey [41]. Ijspeert's work concentrates on improving the parameters of this artificial model, so that it demonstrates performance closer to that of the biological lamprey. Further work on this subject was undertaken by Jimmy Or in his thesis [42] who took Ijspeert's work a step further, developing robust and resilient controllers that can withstand faults or changes in body parameters.

Much of the background presented here is covered in greater depth in the background chapters of Ijspeert's thesis, specifically pages 13-64 [40] and can be considered an excellent reference for this chapter.

3.2 Animal Locomotion and the Argument for Central Control

The motor organisation in even the most basic of animals can be extremely complex and certainly incredibly varied with multitudes of different creatures, all exhibiting different locomotive actions, or *gaits*. It is therefore difficult to imagine that there could be a general principle that would underlie all of them, however it can be seen that almost all locomotion consists of carefully sequenced, rhythmic activity controlled by an underlying neuronal circuitry.

Around the end of the 19th century, British neurophysiologists Sherring and Brown observed that alternating leg movements could still be produced in animals with severed spinal cords

(reported in [41]). From this they correctly deduced that the brain does not directly control the rhythmic behaviour visible in locomotion and that some other mechanism, possibly located in the spinal cord, is responsible.

Since then the nature of this mechanism has been debated. Historically there have been two competing theories for the underlying control of activity that generates this behaviour. Delcoymn, in his 1980 article [43], reviews the arguments for these hypotheses:

The first of these, peripheral control, proposes that the rhythmic movement is a direct function of sensory feedback. That the rhythmic motion is generated through a two stage process whereby the first stage generates the correct timing activity from sensory input and passes this onto a second stage in order to activate the muscles appropriately. However this theory implies that the loss of sensory input would therefore prohibit the rhythmic activity, which more recently was proven not to be the case. (see [43])

The second and now widely accepted hypothesis is that the rhythmic activity is generated by a central mechanism that will operate even in the absence of sensory feedback. Comprising a network of neurons, this network operates as an oscillator producing the correct rhythmic timing activity to the muscles. However these two theory's are not mutually exclusive. Although sensory input is not the primary mechanism, it does contribute to the network as a feedback mechanism. This neural controller is known as a *Central Pattern Generator (CPG)*

It can be seen that the brain plays a very limited role in the coordination and timing of even very complex locomotion. Experiments in the 1960s on a decebriated cat (reported in [41]) showed that a walking gait could be achieved by applying a stimulus to the brainstem. More importantly though, by increasing the amplitude of this stimulus the velocity of locomotion increases, moving from walking gait to a trot and then on to a gallop. Each of these different gaits requires a significantly different motor timing and coordination yet the mechanism for control, now know as the CPG, is able to coordinate the transition between these gaits automatically. Obviously the CPG for a mammal such as this will be significantly more complex than that of the lamprey, but the principle of central control is maintained.

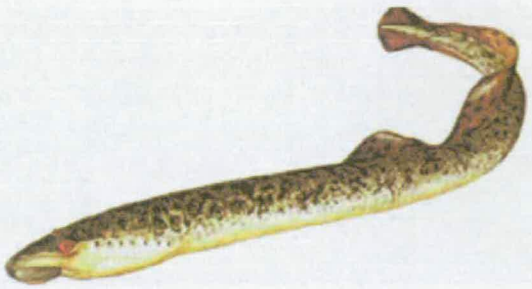


Figure 3.1: *Sea lamprey.* ©*Penobscot River Restoration Project, Maine, USA*

3.3 The Lamprey

The lamprey (figure 3.1) is a jawless, eel-like fish belonging to the family *Petromyzontiformes*. It is a primitive vertebrate that diverged from the main evolutionary line around 450 million years ago before fish appeared and has evolved relatively little since then. It has the basic vertebrate biological structure but features comparatively fewer nerve cells of each type.

Although restricted to colder waters, the lamprey can be found in many locations around the world. In Europe the lamprey is considered somewhat a delicacy, and mainly through water pollution and over-fishing its population on that side of the Atlantic has plummeted. This is not the case in north America however where its populations have soared. There its predatory nature has decimated populations of local fish in the great lakes area and is now considered a serious pest.

The lamprey propels itself by propagating a standing wave along the length of its body; it is this movement that we find so interesting. The locomotion of the lamprey has been the subject of extensive study by neurobiologists looking to understand the underlying methods of locomotive control.

3.3.1 Structure

Another reason the lamprey is used as subject for such extensive study is that its central pattern generator features a large and simple neural structure. The physiological structure is covered in depth in Rovainen's work [44] and a more recent overview can be found by Grillner [45]. As discussed earlier it has been shown that locomotion is not directly controlled by the brain, but rather by a separate local neural circuitry known as a central pattern generator. The lamprey

CPG is remarkably well studied and is located within the spinal cord. It comprises relatively large neurons that have facilitated investigations into its neurophysiological structure. It can be seen [46] that the CPG consists of around 100 individual simple "segments" (see fig 3.2) that control the muscle activation on opposite sides of the body. These segments are then linked longitudinally together so as to coordinate movement along the length of the body.

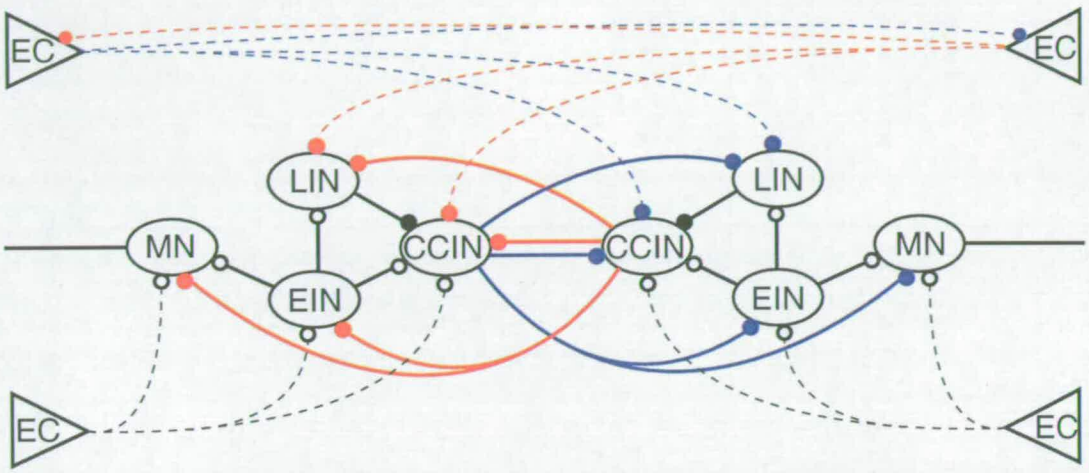


Figure 3.2: *Single segment of lamprey CPG, from [6, 47]. One of approximately a hundred segments that control locomotion in the lamprey CPG. The symmetrical cross coupled nature of the segment is emphasised by the red and blue connections. It can be seen that the connections that extend to the opposite side of the segment are all inhibitory, ensuring that only one side can be active at any time.*

The neurons within this segment have distinct roles. Buchanan and Grillner [48] proposed the circuitry shown in fig 3.2 and showed that it comprises four separate neuron types:

- Motorneuron (MN)
- Excitatory Interneuron (EIN)
- Lateral Interneuron (LIN)
- Contra-laterally and Caudally projecting Interneuron (CCIN)

In addition there is also the Edge Cell (EC). This is a stretch sensitive cell which is located on either side of the spinal cord that contributes to sensory feedback.

Each segment performs as an oscillator, with the frequency being determined by the level of excitation received from the brainstem (or from a excitatory bath if in vitro). This oscillatory behaviour is in part determined by the CCIN neuron and also in part by an undetermined burst terminating mechanism. Once one side of the segment receives slightly more activity than the other, the more active CCIN neuron will inhibit the opposite side, therefore allowing activity to build on that side only. After some time this activity then decays due to a burst terminating mechanism and reverses the inhibition on the central pair of CCIN neurons, thus allowing the opposite side of the network to increase in activity. This contra-lateral activity will continue as long as a base excitation is maintained to the segment, with this level of base excitation (oscillatory frequency) being related to the amount of excitation provided.

3.3.2 Anguiliform Locomotion

The swimming motion of the lamprey is termed Anguiliform and can still be observed when an isolated spinal cord is immersed in an excitatory bath. This is known as *Fictive Swimming* and illustrates that the swimming motion is derived from the neural structure present within the spinal cord. The activity of the spinal cord, and hence the velocity of swimming can be adjusted by altering the overall excitation [49] by adding neurotransmitter agonists such as amino acids or L-DOPA [44] to the bath.

This forward motion is produced by a caudally (rearwards) propagating wave along the body (see fig 3.3). In order to produce this each segment must communicate with the next to generate a delay between activations so that the contraction propagates along the body. Interestingly it can be seen that for a 100 segment spinal cord this delay is maintained at approximately 1% of the period of oscillation. Thus a single wavelength is maintained along the body, independent of the swimming velocity. This activity is also demonstrated in isolated segments of spinal cord with as few as 3 segments.

The significance of this inter-segmental phase delay can be illustrated: as the observed frequency of oscillation of the lamprey is in the region of 0.25 to 10Hz (Reported in [50]) and a phase delay is maintained at 1% per segment, the actual time delay between segments will thus vary 40-fold. This variation is far too large to be explained through simple propagation delay and it has thus been shown that this activity can only be explained by a caudal and rostral inter-segmental coupling along the spinal cord [51, 52].

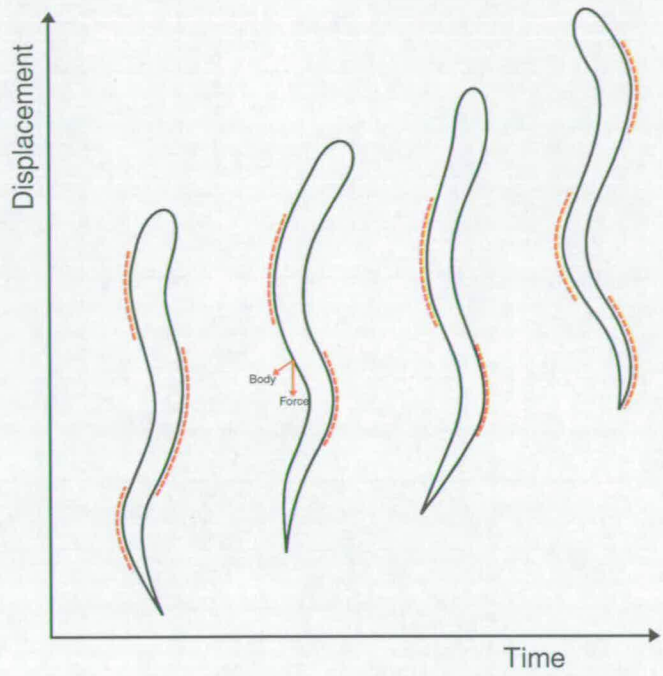


Figure 3.3: *Anguiliform locomotion of the lamprey. Areas of muscle contraction are highlighted. The propagation of a travelling wave along the length of the body can be seen clearly.*

3.3.3 Sensory Feedback

As with any creature, lampreys coordinate movement through the use of sensory feedback. In the lamprey this sensory input is generated by stretch sensitive cells along the length of the body. These stretch sensitive cells, often termed "edge cells" are located either side of the lamprey's spinal cord and have been shown to provide a local feedback connection into the individual segmental oscillators [46]. The sensory input provides a burst terminating mechanism, similar to that described earlier, that controls the intra-segmental oscillation. The action of the sensory feedback can be demonstrated [53] by physically bending the spinal cord and observing that the segments will phase-lock into the same wavelength and frequency as the movement (see fig 3.4). This phase locking ability can be seen if the mechanical actuation is above or below the "resting"[1] network oscillation.

[1]The 'resting" frequency is that imparted to the network by the in-vitro excitatory bath and can vary depending upon this excitation. It does not not imply stationary, but rather a a steady state oscillation.

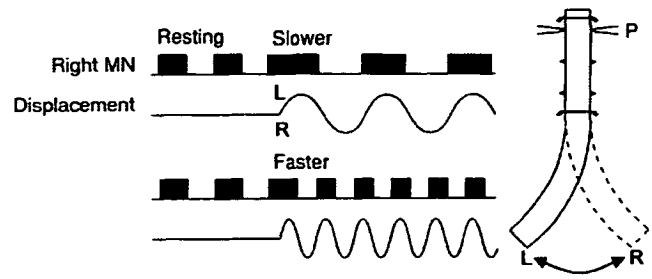


Figure 3.4: *Sensory entrainment in a lamprey spinal cord. Right segmental motorneuron output is shown measured at P. It can be seen that the MN output phase locks to the applied displacement at faster and slower periods than resting. Reproduced with kind permission © S Grillner [46]*

3.4 Computational Models

In order to further examine the underlying circuitry that makes up the CPG it is convenient to produce computational models of the neuronal circuitry. These have been developed by a number of people and use different techniques. The CPG has been modelled in three ways:

Biological models comprise relatively accurate neuron models, using Hodgkin-Huxley type neurons equipped with ion channels [47, 54]. It can be seen that these models have been able to reproduce the majority of observations during fictive swimming experiments. These models have reduced the inherent complexity of these networks by representing populations of neurons as single neuron units. However the complex nature of this network requires many estimated and biologically measured neural properties.

The second method is to use a connectionist, ANN based model that features much simpler models of the individual neurons. By using simpler, generalised models of the neurons it frees us to focus on the synaptic interconnections between neurons and how they generate the physiological responses. Ekeberg and Ijspeert used this method and their models show that the rhythmic activity observed can be reproduced simply through synaptic connectivity [6, 55].

The third alternative is to consider the neural oscillator on a more abstract level. If we accept the rhythmic oscillatory nature of the CPG then it is possible to model the segments as a series of coupled oscillators [56]. This mathematical abstraction is used essentially in order to study the inter-segmental coupling without considering the structure of individual segmental oscillators.

However, for the purposes of this thesis, we shall be considering only the connectionist model of the lamprey CPG.

3.4.1 Connectionist Model of the Lamprey

As previously described, connectionist models are concerned primarily with the interconnections between neurons, rather than the detailed membrane activity and ion flow that occurs in biophysical models. This simplified model is similar in some respects to dynamical recurrent neural networks used within the ANN community (DRNN). Work on connectionist models of the lamprey has been carried out most notably by Ekeberg in 1993 [6] with further study by Ijspeert [40] although other examples can be seen [50, 55, 57].

Ekeberg's model, published in 1993 [6] examines a mechanical and neuronal model of the lamprey that exhibits many of the traits observed in nature. This model uses simpler neurons that were developed to feature much of the richness found in more complex biological neuron models [54] such as frequency adaptation. Each of these neurons is regarded as a population of functionally similar neurons and the mean firing rate, rather than a spiking action potential, is used.

The synaptic connections in this model were hand crafted to produce behaviour that was a good match to the lamprey. In his doctoral thesis, however, Ijspeert [40] took Ekeberg's basic neural model and applied an evolutionary technique in order to develop better interconnections. This method produced a much better physiological match to the lamprey characteristics.

3.4.2 Neuron Model

As we have seen above, the lamprey CPG comprises four different types of neurons. These different neurons, LIN, EIN, CCIN and MN are modelled using a standard leaky integrator model with saturating transfer function. They are described by the equations 3.3 and 3.4. These equations do not model individual neurons, as this would require a more complex model, rather they model a population of neurons of the same type, such that the output, u represents the *average* firing frequency. The neurons are also modelled to include a frequency adaptation term, shown as τ_A which results in a slight decrease of the firing rate over time. Its effects are shown to affect the inter-segmental coordination rather than the segmental oscillation frequency (although τ_A may indirectly have some impact on this).

The neuron parameters used by Ekeberg and Ijspeert are given in table 3.4.2.

Neuron Type	Θ	Γ	τ_D	μ	τ_A
EIN	-0.2	1.8	30ms	0.3	400ms
CCIN	0.5	1.0	20ms	0.3	200ms
LIN	8.0	0.5	50ms	0.0	–
MN	0.1	0.3	20ms	0.0	–

Table 3.1: *Ekeberg & Ijspeert's' neural parameters. [6] Θ is the threshold, Γ is the gain and τ_D is the time constant. μ and τ_A are the coefficient and time constant of frequency adaptation respectively.*

$$\dot{\xi}_+ = \frac{1}{\tau_D}\left(\sum_{i\epsilon\Psi_+} u_i w_i - \xi_+\right) \tag{3.1}$$

$$\dot{\xi}_- = \frac{1}{\tau_D}\left(\sum_{i\epsilon\Psi_-} u_i w_i - \xi_-\right) \tag{3.2}$$

$$\dot{\vartheta} = \frac{1}{\tau_A}(u - \vartheta) \tag{3.3}$$

$$u = \left\{ \begin{array}{ll} 1 - \exp\{(\Theta - \xi_+)\Gamma\} - \xi_- - \mu\vartheta & (u > 0) \\ 0 & (u \leq 0) \end{array} \right\} \tag{3.4}$$

The mean firing frequency given by u includes stimulation by the pre-synaptic inhibitory $\dot{\xi}_-$ and excitatory $\dot{\xi}_+$ neurons. These are in turn given as the sum of the output of the post-synaptic neuron[2] u_i and the associated weight to the pre-synaptic input, w_i.

3.4.3 Genetic Optimisation

Ekeberg's artificial model [6] featured hand-tuned synaptic weights, developed through measurement and trial and error to produce similar activity to the biological model. Alternatively Ijspeert's model used a genetic optimisation routine to develop the synaptic weights and his model can be seen to show better and more realistic results than Ekeberg's.

Genetic Algorithms (GA's) are very powerful tools and ideally suited to optimisation of problems featuring a large search space such as neural networks. A genetic algorithm encodes each possible solution as a string and by using genetic operators such as reproduction, survival and mutation, evolves these solutions to produce a best fit to the problem. The key to good use

[2]This neuron representation is defi ned as a population of appropriate excitatory Ψ_+ or inhibitory Ψ_- neurons.

of a GA is to define the evaluation function that will decide how "fit" a solution is and thus which 'solution strings' are allowed to mate and propagate to subsequent generations. Possibly due the fact that the solution can be easily represented as a series of synaptic weights or due to their efficacy with large search areas, genetic algorithms are a common choice for optimisation of neural networks.

In the case of the lamprey CPG, we will use a GA to develop the synaptic weights in a similar way to Ijspeert. We will further discuss the use of GA's in section 4.4 and show how this technique is ideally suited for use in this thesis.

3.5 An Engineering Approach

We have discussed the neural structure of the lamprey and have looked at the features of CPG, however now we need to see how we can apply this model to the application of WEC control.

If we consider a simple point absorber WEC as discussed in section 2.5, we know that for optimal control using a latching strategy (see section 2.4.2) there are optimal lock and release points during the wave cycle. The timing for these points varies as the wave period changes. The calculation of the optimal release point is difficult and requires some knowledge of the future wave. However, if we dismiss *optimal* control and instead consider *practical*[3] control we can still gain significant power increases. Therefore if we can approximate the correct unlatch points from the current device displacement, we can produce a simple and effective control system. The lamprey CPG described here provides the inspiration for this idea.

If we consider the sensory entrainment effect illustrated in figure 3.4 we can see that the physical bending of one part of the body will cause the whole network to conform to the the phase and frequency of the bending. This is due to the sensory feedback and inter-segmental connections along the length of the spinal cord. If we now consider a single segment, the inter-segmental coordination can therefore be neglected. The system now becomes a simple oscillator that will entrain to the excitation frequency present at the sensory input. This suggests that if the synaptic weights of this network are adjusted sufficiently while still maintaining the key architecture, the phase and the response of the output, relative to the sensory input, can be modified.

[3]By practical control we are referring to a non-optimal method that may be implemented simply and easily while still offering a signifi cant improvement over an uncontrolled system.

If we therefore take this single segment and apply the current device displacement as the sensory input, the neural oscillator would then be able to adapt to the same frequency as the device. Although the frequency will match the entrained frequency, the phase and response of the segment are thus dependent upon the synaptic connectivity. Therefore the segment would perform as a tunable, adaptable control loop, which once optimised could then be used to trigger the correct unlatching points in the described system.

The use of the lamprey CPG as a control mechanism can also be seen if we look at an analogue such as the articulated wave energy converter *PELAMIS*, developed by Ocean Power Delivery [5, 28]. This device consists of four floating cylinders, articulated at their ends via hydraulic rams. As the wave passes under the device, the cylinders move relative to each other, pumping hydraulic fluid and generating power. It can be seen that this wave induced motion is quite similar to the anguiliform swimming of the lamprey. Also, as the wave passes down the length of the device the joints also exhibit similar phase delays as those seen in the lamprey.

At this stage it may help to re-state the original hypothesis that:

Effective control implementation for Wave Energy Devices can be developed from (neuro)biological exemplars

Thus by identifying the lamprey as a suitable neurobiological system we can develop the single neural segment to act as a controller for a single buoy. Furthermore, by developing this model it may be possible to add additional segments and develop inter-segmental connections to produce an effective control system for a complex articulated device similar to the *PELAMIS*.

3.6 Chapter Summary

This chapter has presented the concept of central control and has explained how the locomotion seen in vertebrates can be explained through the principle of a central pattern generator. This principle has been expounded through the use of the lamprey neurophysiological structure, a vertebrate with a basic neural structure that has been the subject of extensive research.

Although it is known that locomotion is a complex task, it can be seen that the brain does not deal with this coordination. Instead it is controlled by local subsystem, the central pattern generator and it can be seen that the same principle is featured in all animals. Although in general larger creatures have more complex gaits and an appropriately more complex CPG, the underlying principle still remains and a good understanding can be accomplished by examining smaller, simpler vertebrates, such as the lamprey.

The biological structure of the lamprey CPG is now well known, it has been the subject of extensive study for decades, however much of this knowledge has been gained though numerical modelling. Complex, biologically accurate neurons which have been simulated have exhibited most of the activity observed in in-vitro experiments.

As a means to understand the full body actions and investigate the inter-segmental phase delay mechanisms, the field of Artificial Intelligence has provided invaluable techniques. By influencing the creation of much simpler neurons, the interconnections between neurons could be studied in more depth and it has been shown that the rhythmic behaviour can be attributed to these synaptic interconnections.

Knowing the significance of the synaptic interconnectivity, the lamprey CPG can be seen to be quite complex, even assuming the inherent symetricity a single controller can consist of up to 64 connections. Considering the number of different possible structures, Ijspeert artificially evolved controllers that would exhibit similar activity to their biological counterpart. These artificial CPG's exhibited much structural similarity to the lamprey, but were found to represent the natural locomotive activity more accurately than previous models.

By using simplified representations of this biological network, we seek to develop a controller that is designed to fit the control needs of a very different application. By using the same evolutionary optimisation techniques Ijspeert used to produce very successful biological controllers, we can seek to produce different, but equally successful controllers; ones that will

use the unique characteristics of the lamprey CPG to implement effective latching control to a point absorbing wave energy device.

Chapter 4
Tools and Methods

This chapter introduces the methods and techniques used in this thesis and goes some way to justifying their inclusion. Starting with an expansion of the neural control method touched upon at the end of chapter 3 we look at how this may be implemented practically and the simplifications required to do so. A computer based model of the dynamical systems so far introduced is presented along with a description of the methodology and processes involved. Following this, implementation and optimisation techniques for the control system are discussed and their applicability to the problem is illustrated along with some more advanced methods that may be used at later stages. Finally, problems of scale associated with the optimisation process are discussed and the inclusion of a multi processing environment is presented.

4.1 Introduction

Chapter 3 introduced the concept of using a biological neural system as a controller for a wave energy converter. By using the basic neuronal structure from the lamprey CPG and applying it as a control loop to a simple single degree of freedom WEC it is intended to implement a latching control strategy.

The lamprey central pattern generator is a biological neuronal network that provides locomotion. At the segmental level its structure comprises 8 neurons divided into two symmetrical units and can be seen to operate as an oscillator. The characteristics of this oscillator are dependent upon the interconnections between neurons and, although the biological synaptic structure is well known, it is unlikely that this structure will be appropriate for this proposed application. This chapter will outline techniques and methods that can be used to tune the synaptic weights to fit the suggested control application.

4.2 Implementation of Neural Control

In order to implement control to a system we must define the inputs and outputs of the system and define the information that is required. In the WEC, a simple latching strategy will be used as the method of control. Section 2.4.2 describes how this operates, but in essence it locks the body at the extremes of displacement and then releases at some point later (determined by the control) in order to achieve a quasi-resonance with the exciting wave force. As this can be implemented very straightforwardly, and requires only a single control parameter, latch time, it was seen to be ideal in this instance. It is known that latching strategies have many disadvantages[1], however latching is a cycle by cycle control method and in this application its simplicity more than makes up for these drawbacks.

Although mathematically simple, in reality, latching can present considerable challenges in implementation. One theoretically plausible method would be to incorporate it within a hydraulic power take-out system. Assuming a rigid tether, a valve in the system would stop flow and thus lock the device in its current position. This method of hydraulic locking could potentially be implemented into an articulated system such as the PELAMIS. However, as mentioned in section 2.6, locking a real device at its extremities would induce additional large forces and would involve some engineering challenges. Although latching control has been demonstrated physically at model scale, employing latching control on a full scale device is somewhat unrealistic. Nevertheless, this is a proof of concept study and concentrates on the neural application of WEC control and although it investigates latching in a number of configurations its focus is upon the *application* of a control method rather than the actual method of control used.

Figure 4.1 shows the method of integration of the two systems. The physical displacement of the body in the water is directly mapped to the EC input of the network. In order to maintain the contra-lateral operation of the network positive displacement is considered as left side excitation and negative displacement as right side excitation. Although we consider the physical motion in a vertical plane, we maintain the terminology used for the CPG network, meaning that where we refer to left and right output from the network, this translates (through 90° rotation) to up and down motions of the buoy. The scale of this mapping is not critical as the actual values passed to the neurons are scaled by the magnitude of the EC synaptic weights.

[1]These can include potentially excessive accelerations at lock/release, the physical implementation of a latching mechanism and a non-continuous power output.

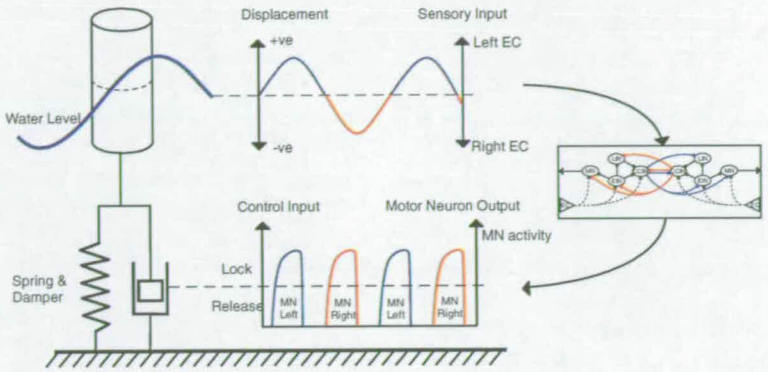


Figure 4.1: *The control loop between the WEC and the lamprey CPG single segment. The translation between the physical displacement and the EC input can be seen in the upper graph, while the lower graph shows the MN output and how it can be used to trigger the locking/releasing of the device.*

The reverse translation, from the motor neuron output to the WEC is important as this will determine the control strategy applied. For a latching strategy, as we have discussed, we need only to define the release point or the total latch time. The relationship between MN output and device control is shown in figure 4.1 in the lower graph. In order to translate this MN output into a latching strategy, we can use either the time the MN output pulse is above a threshold to define the total latch time or the rising/falling of the MN pulse past a threshold to define an absolute release point. Either of these methods are valid, but for simplicity the latch release point is defined as the falling edge of the MN pulse past a pre-defined threshold.

4.2.1 Simplifications

If we examine the requirements for our control system it is possible to make certain simplifications:

The lamprey-based neuronal network we are using is very straightforward, to control a single buoy it requires only eight neurons for operation, and the individual neurons are represented by a very simplified model of their biological counterparts [6]. However, the system becomes complex in the representation of the synaptic weights. If these eight neurons are allowed to form connections to all the neurons in the system, both inhibitory and excitatory, this produces a matrix of 128 values. Combine this with the synaptic connections from the edge cells (inhibitory only - 8) and brainstem (excitatory only - 8) and it gives 144 separate weights.

As we have no clear idea of what synaptic connectivity is required for the segment to operate in the manner that we require it is difficult at this stage to eliminate any connections.

In order to reduce this complexity and facilitate simulation of the neural network, Ijspeert [40] used the inherent symetricity in the lamprey CPG to enable simplification of the model. As we are aware, the oscillatory nature of the neural network is due in part to the symmetrical arrangement of neurons. Therefore, by simply mirroring connections from one side to the other the number of connections is reduced significantly. Another simplification Ijspeert used was to limit sensory input to the same side as it originated, rather than the contra-lateral connections that can be seen in Grillner & Ekebergs [6, 47] description of the neuronal circuitry. The resulting simplified network can now be reduced to 72 interconnections.

4.3 Simulation

As has been discussed, the overall model comprises of two systems, a connectionist neuronal network and a mechanical WEC model. Independently these systems can be modelled quite straightforwardly using the equations already presented. Both systems are time dependent and rely upon solutions to differential equations.

4.3.1 WEC Simulation

Initially developed and verified using MATLAB, the final model was implemented in the C programming language [58] and uses a variable step Runge-Kutta differential equation solver derived from one published within "Numerical Recipes in C" [59][2][3]. One consideration with the mechanical model was that it should be able to respond to a random or user-supplied stimulus and it should not depend upon, or require, future knowledge of the wave. In fact future knowledge is required to calculate the frequency dependent hydrodynamic properties of the device. However as we would be applying control forces to the device this would very much complicate the dynamics of the system. For simplicity, we have therefore assumed the hydrodynamic properties to be static values (see section 2.11). Although it is understood that this is strictly not accurate, it can be considered a rough approximation and adequate for the

[2]This routine has since been highly modified to fit this particular application but retains the structure of the original.

[3]It was found that during simulation the C code ran at least 10 times faster than the MATLAB equivalent.

purposes of this thesis.

4.3.2 Neural Simulation

Initially developed in MATLAB and then implemented in C, a neural model of the lamprey CPG was developed using Ekeberg & Ijspeert's neural parameters. Modelled using a simple fixed step Runge-Kutta differential equation solver,[4] this model performed as indicated in the literature [40] and was verified with a 100 segment lamprey model.

4.3.3 Combining Systems

Both the mechanical WEC model and the neural network model are dependent upon each other and in order to implement the presented control method, the information interchange between these elements is critical. Ideally data would be interchanged between systems at every time-step, but because these systems have been implemented and verified as separate components, the time overhead to re-write them into a single ODE loop was too great. This, however, can be compensated for somewhat by implementing an overall time-step ("chunksize"), that is a magnitude greater than the individual system time-step and interchanging data after each chunk.

4.3.4 Implementation

The operation of the overall system is shown in figure 4.2. The initial data is divided into a predetermined number of chunks and processed one at a time. The buoyancy calculations are solved for one chunk, then the final value of displacement and velocity is converted to the EC value for the neural simulation. This is then also solved over the same time chunk and the resulting final MN value converted into a damping value to be used for the mechanical simulation in the next chunk. A more detailed description of the operation can be found in appendix A.

[4] A variable step solution was attempted, but not implemented due to time restrictions. Ultimately it was decided that the fi xed-step solution was an adequate trade-off between accuracy and speed.

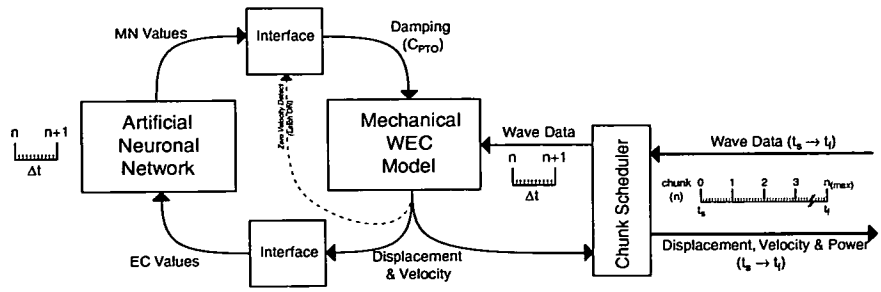


Figure 4.2: *Block diagram of overall simulator operation.*

4.4 Evolutionary Algorithms

At this stage, it is clear that we know the form of the synaptic connectivity, but not the weight values that are required for optimal, or even usable, control. Therefore it is necessary to employ a technique that will enable the different possibilities to be analysed. Ijspeert [40] employed a evolutionary technique that is well suited in this instance; he used this to develop CPG's that were proved to be a very close physiological match to the biological lamprey. This evolutionary technique, sometimes known as the genetic algorithm, is particularly applicable to the optimisation of neural networks.

4.4.1 Genetic Algorithm

The genetic algorithm (GA) is a stochastic search routine that locates optima using processes similar to those in natural selection and genetics. Individual solutions are encoded as 'solution strings' and these are manipulated in a biological analogue, using processes such as selection, mating, mutation and migration. Hence by applying the principle of 'survival of the fittest' the GA produces better and better solutions [60]. This results in a heuristic search method that can effectively navigate extremely large search spaces. If correctly designed it can be relatively immune to local minima (due to variation [mutation]) and will always produce good, although not necessarily optimal, solutions.

As mentioned above, individual solutions need to be encoded into a structure. This structure is a long string comprising the individual elements of the solution. The mapping of the solution into the string can be done using an appropriate alphabet, such as binary, real-value, integer, ternary, etc, although binary and real values are by far the most common. For the problem at hand, a real valued representation is used, mapping each synaptic weight as a value that makes

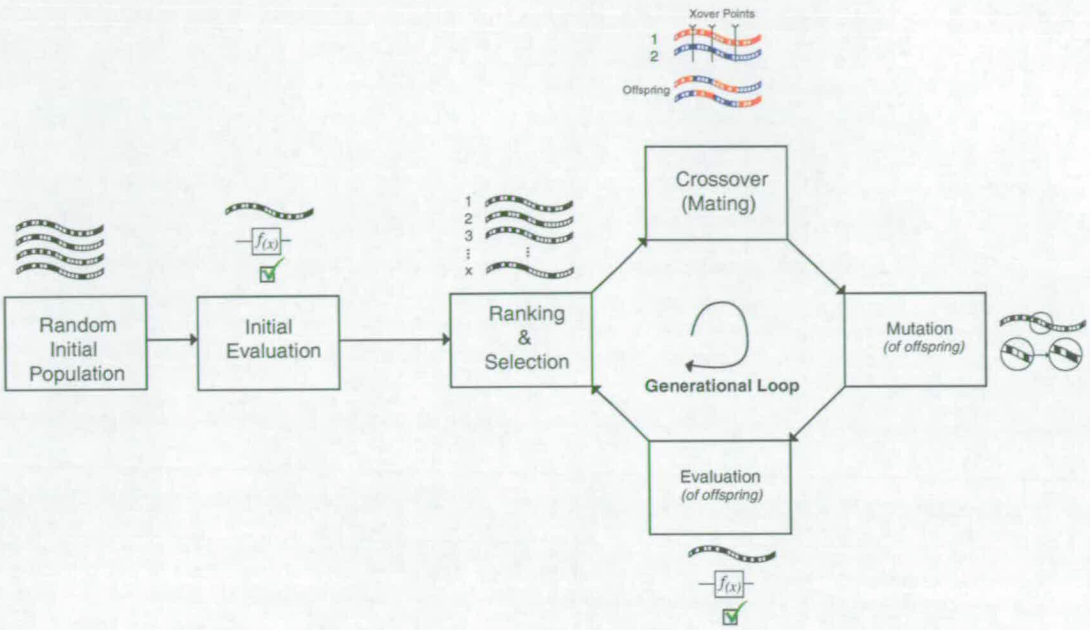


Figure 4.3: *Basic operation of the evolutionary algorithm.*

up the solution string. In addition there are many advantages of real-valued populations over more common binary encodings, as shown by Wright[61].

Once the representation has been determined, the evolutionary process is quite basic and comprehensible. Figure 4.3 shows the major blocks in a basic genetic algorithm. Initially a population of strings are generated with random collection of values, these are then evaluated for fitness before entering the main generational loop. The fitness function is the most critical element of the GA as this dictates how good a solution is. Once evaluated the strings are ranked according to their fitness, with the most fit individuals being selected for breeding. Breeding, also known as crossover or more accurately, recombination, involves the exchange of values between two individual solution strings to create two new solution strings (offspring). The simplest operator for this function is knows as single point crossover which simply exchanges data about a randomly chosen point within both strings, although variations such as double point, multi-point, line, shuffle, intermediate and reduced surrogate can be used to provide a better distribution of the individual values within each string [62–65]. Real-valued populations can provide complexities when dealing with recombination, however variations on the basic binary operations may be applied to ensure that only valid values are produced as a result [66].

In the GA presented here, the crossover method used is intermediate recombination as this is more suited to a real valued population.

Following selection, strings are mutated in order to ensure that the probability of covering any area of the search space is never zero. In a binary representation mutation normally takes the form of selecting bits randomly with a very low probability and flipping the value of a single bit at this point. With real-valued populations mutation is achieved by perturbing a random selection of values within the solution string with a probability higher than that used with a binary representation [61, 66, 67].

The new individuals are then evaluated, ranked and re-inserted into the original population. In order to keep the overall population at a constant size the new individuals will replace the old. A common effective technique is the *elitist* strategy, where the best individuals from the previous generation are allowed to propagate through to the next generation with new individuals replacing all other individuals in the previous population. In addition, another successful strategy ensures that the oldest members of a population will be replaced, thus being in keeping with a biological scheme whereby eventually every member of the population will be replaced [68].

The above steps are repeated until the termination of the algorithm. However it is not always clear when a maximum has been found. Due to the nature of the method, the maximum fitness may remain static for a number of generations before a better individual is found. This can be a particular problem in applications such as ours as we do not know what the optimal value will be. In this situation the algorithm is terminated after a fixed number of generations, the number of which is determined through a compromise between time taken and improvement seen.

The field of genetic algorithms is vast and much work has been done to develop new techniques and better operators for more complex problems. However the basic operators remain suitable for many applications such as this. A large number of advances deal with improving convergence time, which is considered one important benchmark when dealing with GA performance.

4.4.2 Advanced Methods

For more difficult problems there are variations on the basic GA that can be employed. One important method is the introduction of multiple populations, which has been shown

in most cases to improve the performance of a single population method [69, 70]. Also known as the Island or Migration model, each of the individual *sub-populations* are evolved independently for a number of generations after which individuals are allowed to migrate between sub-populations in order to increase genetic diversity. Figure 4.4 shows how a ring based model would work, with individuals being passed between sub-populations as shown. In the multi-population GA used later in this thesis a fitness based ring model is used, whereby the most fit individuals are migrated to the adjacent sub-population after a fixed number of generations.

A technique that can be beneficial in complex problems is the idea of "non-redundant selection" whereby, after recombination, identical solutions are removed and replaced with either a different solution string (that would not otherwise have been 'fit' enough to be included) or a random string. Illustrated by Zhang [71] in his algorithm, it can be shown to increase the navigability of the search space in particularly difficult problems.

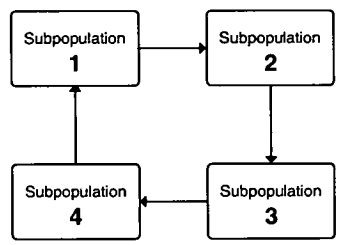


Figure 4.4: *A ring based multi-population GA. Individuals are migrated between sub-populations in the manner shown.*

4.4.3 Implementation

The purpose of the GA in this application is to determine the best synaptic weights for a neural network so it may effectively apply a latching strategy to a buoy excited with an incident waveform. This synaptic optimisation is done off-line with various incident waveforms, developing a set of synaptic weights that can be used as on-line control (in pseudo real-time).

The genetic algorithm used here is implemented in MATLAB, using the GA Toolbox [72] developed by the University of Sheffield. This toolbox provides a comprehensive set of genetic operators that can be used to quickly and efficiently develop advanced genetic optimisation routines. A full explanation of the GA's used in this thesis is provided in Appendix B.

It was found quite quickly that while MATLAB proved perfectly adequate in terms of speed of operation to run the GA, the evaluation of the individuals was a very computationally intensive task. The simulation of the neural network along with the mechanical interaction of the buoy for a period of 60 seconds, running on a single 2.8GHz Pentium 4 processor takes approximately 30 seconds[5]. Assuming a GA population of 40 individuals, it takes 20 minutes to evaluate a single generation[6] and in practice a 500 generation evolution takes around 4-5 days. Therefore it is clear that times for evaluating a reasonable number of generations on a single processor is excessive.

It can be seen that genetic algorithms are an ideal candidate to take advantage of parallel processing. If the algorithm is run on one processor, if correctly written, each string can be evaluated as a separate instance, which can provide a linear speed-up of the evaluation proportional to the number of processors in the system[7]. In order to take advantage of multiple processors an efficient method of message passing between processors is required and in general this is dependent upon the system hardware employed.

The school of Engineering and Electronics at the University of Edinburgh provided access to a farm of 20 2.8GHz Pentium 4 linux machines. These machines are stand alone and do not exist in any kind of Beowulf [73] cluster environment, each requiring an SSH connection in order to communicate. In order to use these machines in a parallel environment involved the adaptation of a Perl script, tfSSHServer.pl[8] that was able to implement the appropriate message passing. This script opens an SSH tunnel to a number of machines and then passes the appropriate startup instruction. As these machines use NFS[9] each program instance writes the result to a common file area, which is monitored and collected by the algorithm. This multiprocessor operation resulted in a very significant performance performance increase, nevertheless, with the more complex models presented towards the end of chapter 6 a 1000 generation simulation, using a ring-based multi-population genetic algorithm, takes around 3-4 days to complete.

A full description of the operation and the code used can be found in Appendix B.

[5]This is after the improvements of coding the simulator in C. Previous MATLAB code was taking around 2-3 minutes for the same simulation

[6]These simulation times assume the simplest mechanical model (single buoy, refer to figure 2.5) and can only increase with mechanical complexity.

[7]Each instance can be treated as a separate program that can be run independently.

[8]Developed by Pete Ottery within Informatics at Edinburgh University.

[9]Network File System. Unix (& Linux) system whereby the user can access the same file system no-matter which machine he logs into.

4.5 Chapter Summary

This chapter can be seen as an introduction to the methods employed, and not as an in-depth comprehensive coverage. These methods will be clarified as they are presented in the following chapters.

The whole body of work presented in this document is reliant upon the computer models presented in this chapter. These models have been developed to be as simple as possible in order to reduce the possibility of error. By using a double time step approach the model attempts to solve the neural and mechanical equations simultaneously while treating them as separate programs. This allows quicker integration and compartmentalisation of the two systems, enabling both systems to be run separately for verification or together for evaluation.

The method of using evolutionary algorithms as a search technique to find synaptic weights for the neuronal network is a tried and tested one. Through the use of various genetic operators the algorithm is able to effectively navigate the search space in order to progressively find better and better solutions. The heuristic nature of this means a good solution will always be found, although without the certainty that it is the optimum. Nevertheless, as the algorithm is also based on stochastic methods, by running several separate evolutions and comparing the resulting fittest individuals, the probability of finding a true optimum can be increased[10].

The downside of using a genetic algorithm on a problem such as this is the increased computer resources required to evaluate the individuals each generation. By using multi-processor methods such as the Perl SSH method presented here we can reduce the run-time for the algorithm at the expense of increased computing power. This multi-processor technique, consisting of 20 2.8GHz Pentium 4 processors provides a total processing power of approximately 25GFLOPS[11]. This high-power resource reduces the time taken to run GA by greater than an order of magnitude. This allows us to evaluate individuals using more than one wave condition and offers the possibility of using more advanced GA techniques, such as multi-population or non-redundant search, which can require a larger number of individuals per generation.

[10] If the best individual from a series of separate evolutions is very similar, it is likely that these resulting solutions will be very near the optimum.

[11] The floating point performance of a P4 3.06GHz is approximately 1414MFLOPS [74] (thus 2.8GHz $\approx$ 1293MFLOPS), thus with 20 machines we can achieve a maximum of 25.877GFLOPS.

Chapter 5
Preliminary Results - Combining Mechanical & Biological Systems

5.1 Introduction - Developing A Neural Controller

This chapter will cover the coupling of the mechanical model to the neural model and covers most of the initial and possibly most important results. Starting with a more in-depth look at how we can use the neural oscillator from the lamprey CPG to control a simple point absorbing WEC, we implement a tentative one-way connection between the two systems. This will show that by applying sensory input to the neural network it is able to adapt and phase-lock to the applied waveform. A fully combined mechanical and neural system is then introduced and we detail how we can use a Genetic Algorithm to optimise the neural network for this configuration. The resulting system is tested in a variety of unrealistic monochromatic waves, with more realistic and complex sea-states being investigated in the following chapter. The effect of the variations in the neuron time constant is then investigated and finally the effect of varying the system damping level is described.

5.1.1 Control Structure

Whilst it is clear that the synaptic weights and neural parameters of the biological lamprey segment clearly illustrate its operation as a neural oscillator, it is very likely that adjustments will be necessary to fit this particular application. The biological lamprey features segment oscillation frequencies in the region of 0.5 - 2Hz. However, assuming that we require the lamprey to oscillate at the same frequency as the wave, we need to look at frequencies in the range of 0.08-0.2Hz (5-12 seconds period) which is roughly an order of magnitude lower than the artificial swimming models developed by Ekeberg or Ijspeert. Fortunately it was found that, by increasing the neuron time-constant by a similar magnitude, the network would operate exactly as before but at a lower frequency. Using these modified neurons, the global excitation applied to the network could be increased from 0.3 to 0.8 and resulted in a linear decrease in

period from approximately 10s to 5s. The original and updated neuron parameters are shown in table 5.1.

Neuron Type	τ_D		τ_A	
	orig	*new*	*orig*	*new*
EIN	30	450	400	6000
CCIN	50	300	200	3000
LIN	20	750	-	-
MN	50	300	-	-

Table 5.1: *Modified neuron time-constants in milliseconds. These have been adjusted in order to produce the required oscillation period*

Initial tests involved applying a sinusoidal excitation to the sensory input (EC input) of an oscillating segment and monitoring the effect that this has on the period and phase of the of the MN output. Due to the operation of the network, in normal swimming motion the stimulation applied to the left and right EC input is positive only and the L/R sides are 180° out of phase. Accordingly, the excitation applied to the EC is formatted in a similar manner.

It can be seen that this applied excitation results in the period of the segment motor neuron (MN) output becoming matched to the period of the applied excitation (to the EC input). This effect was only apparent within a small bandwidth of approximately ± 1 second either side of the natural period of the network[1], but demonstrates that appropriate sensory input can be used to affect the phase and frequency of the output.

This simple phase locking ability exhibited here can be taken a step further if the motor neuron output is used to effect a controlled parameter. If this controlled parameter is then monitored and fed back into the sensory input, we can implement a simple control system. With reference to the system of an ocean wave energy converter, the controlled parameter could be a device parameter that can be adjusted to optimise the power. This would therefore directly affect the motion, meaning the buoy displacement, z, for example, can be used as a sensory (feedback) input. The resulting MN output will then be phase-locked to the buoy displacement and if correctly optimised, may be used to control modulation of certain mechanical parameters of the WEC on a cycle by cycle basis.

[1]The natural period of the network, using our modifi ed neuron parameters was 5 seconds and was produced by applying a global excitation of 0.7

It was shown in chapter 2 that the simplest and most appropriate control mechanism for the point absorber WEC shown here is latching control (see section 2.4.2). This can be implemented simply by switching the power take-off damper between two pre-set levels, one very high level that will effectively lock the displacement of the buoy and a lower damping level that allows the buoy to move and generate power. Therefore in order to implement such a system there must be a parameter that triggers the switching between damping levels. For such an appropriate latching strategy (for periods longer than body resonance), two latching (on-off) cycles are needed per cycle and it can be seen that, quite conveniently, the neural segment also produces two output pulses per cycle, generated alternately from the left and right motor-neurons[2]. If we allow the transition between these two damping states to be determined by the MN output, then the MN pulses will control the damping level. Then assuming that as suggested above, once connected in a feedback loop, the MN period will match the wave period. As a result the phase (and shape) of the MN output can be optimised so that the MN pulses occur in-line with the optimum latching transitions and hence produce a valid control strategy.

5.1.2 Optimisation Method

The relationship between the sensory (EC) input and the MN output is determined mostly by the synaptic interconnectivity. Therefore in order to fit the given control scheme this network will need to be optimised. Even with a network of only eight neurons, this creates a complex problem that lends itself well to a genetic search application. If we describe all possible solutions to this problem in terms of binary or real-valued strings, we can consider them to be analogous to biological chromosomes. We can therefore take a population of random chromosomes (solution strings) and by using a survival of the fittest scheme, we can apply genetic operators such as selection, mutation and crossover (mating) and evolve the fittest possible individual.[3]

In order to apply evolutionary optimisation to this problem we must first set out two important criteria: how we define a solution as a string and how we evaluate how "fit" each 'solution string' is. Fortunately neural networks are particularly well suited to being transcribed as strings

[2]Looking at the output of the neural network(MN), it can be seen to consist of a series of pulses of a period and phase determined by the excitation, synaptic connectivity and the sensory input.

[3]Although evolutionary techniques *can* and often do produce an optimal solution, there is no certainty that this *will* happen. What can be said however is that they will always produce a good solution.

as the solution can be described solely in terms of the synaptic weights and these can be easily written as either a binary or real valued vector. In this problem with eight neurons and allowing for a fully connected network (every neuron joined directly to every other neuron) this results in 64 excitatory and 64 inhibitory synapses, plus 8 sensory inputs and 8 global excitation inputs. However, as the network must be symmetrical in order to retain its oscillatory nature, we can fully describe the connectivity with only half of these synaptic weights, this now results in 72 weights (as opposed to 144) which will be represented as a real valued string.

It is necessary to also define the fitness of each solution and this is done with an evaluation function (aka fitness or objective function). This function can be quite complex as it describes which solutions are better that others and guides the evolution in the appropriate direction, avoiding local maxima. For this application, our primary objective is to maximise the power developed in the power take-off damper (C_{PTO}). However, we must also provide incentive to solutions that are progressing in the correct manner. Accordingly, the fitness for each individual is taken as the average power developed over a 60 second simulation. Although, in order to reward correct oscillatory behaviour, we penalise solutions that produce less than 10 latching transitions in a linear manner so that a solution with no latching receives zero fitness. This avoids chromosomes tending towards a fixed damping solution.

The genetic algorithm script was written in MATLAB, using the "GA TOOL-BOX" [72]. A simple elitist algorithm was employed that kept the best 10% of the population each generation. A real-valued representation was used, along with intermediate recombination as the crossover operator and a flat mutation rate was applied with an additional 10x increase in the probability of mutation every 20 generations in order to promote diversity. Full details of the genetic algorithms used can be found in Appendix B.2.

5.1.3 Preliminary Latching

As a simple test of concept to implement a latching strategy straightforwardly, we define a level of the motor-neuron output arbitrarily that will trigger the mechanical model to "latch-on" when above and release when below. In this work this is referred to as a "full latching" strategy.

The controller whose results are shown in figure 5.1 was evolved using a wave period of 6s and was the result of 250 generations of evolution. It can be seen here that the output follows closely the ideal profile as described in 2.4.2. It is also seen that the MN output triggers the

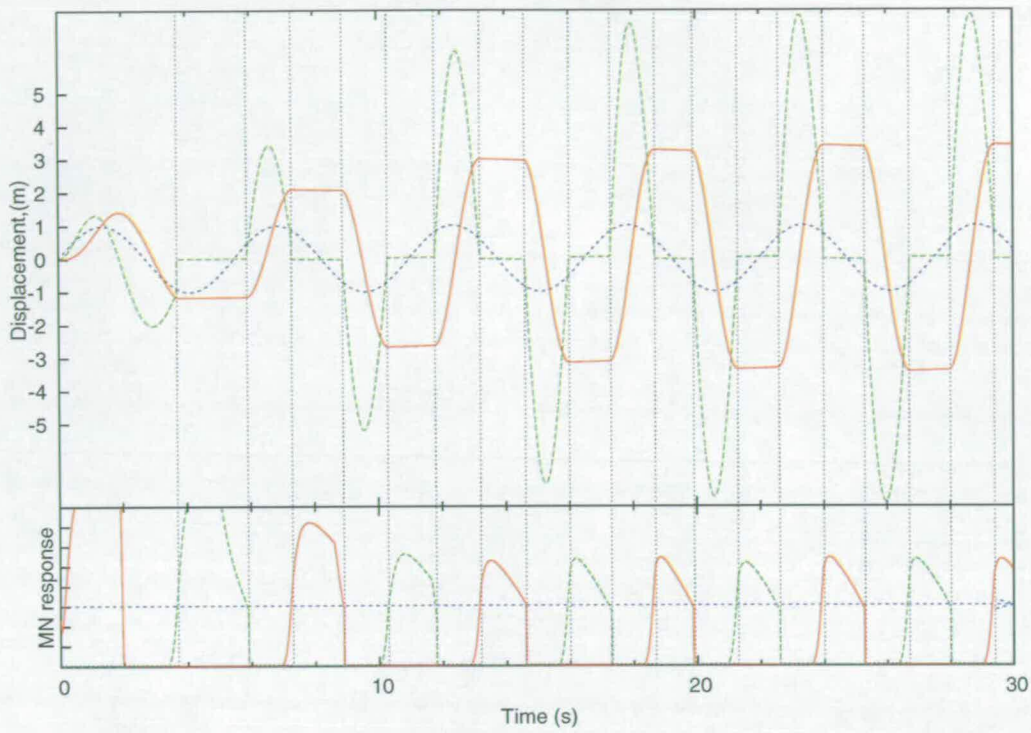


Figure 5.1: *Output of the evolved controller from zero to 30 seconds. The lower trace shows the motor neuron output left (red) and right (green), the latch fix-and-release points are illustrated by the vertical lines. The induced displacement (red) and velocity (green) of the buoy is clearly displayed compared to the incident wave (blue). It can be seen that the displacement of the buoy is much greater than the incident wave height, this is a commonly demonstrated feature of a correctly executed latching strategy.*

"latch-on" points so that they co-incide with points of zero velocity and the resulting output phase is shifted exactly 90° away from the input. As stated before these are known prerequisites for a latching strategy so it is reassuring to find that the solution developed tends towards the theoretical optimum.

5.2 Practical Latching

As we have just stated, it is a prerequisite of latching that the device is fixed only when the velocity becomes zero. It is therefore possible to monitor the velocity of the device and detect when zero velocity occurs. If we take this point to be the ideal "latch-on" point then only the

release point needs to be determined. In regular waves this can be determined theoretically [3], however in this situation we can use the MN threshold specified previously to trigger the release point. This is done by monitoring only the falling edges of the MN output (left or right) with the threshold level arbitrarily set to be 50% of the nominal maximum MN output. Latching (*fixing*) of the device is accomplished by manipulation of the power take-off damper such that the latched state becomes: $C_{PTO(latch-on)} \ggg C_{PTO(latch-off)}$ when $\dot{z} = 0$ [2] conversely the unlatched state becomes: $C_{PTO(latch-off)} = C_{PTOevolved}$[4].

5.2.1 Wide-Bandwidth Evolution

We have shown (see figure 5.1) that for a single frequency monochromatic wave, an evolved neural network is able to correctly determine an accurate latching strategy. This result, although interesting, is of no real use for a system that will work with changing period incident waves, such as a realistic ocean wave. In order to attempt to develop a controller that can implement a latching strategy across a wide bandwidth we must define the wave periods of interest and from chapter 2 we can see that, in general, wave periods range from as low as 4s up to 12s. It can be seen that the mechanical model provides a resonant peak around 2.8 seconds, which is much lower than the wave range of interest. However this is convenient as a latching strategy is ideally used for improving performance at periods longer than resonance (see figure 2.8).

5.2.1.1 A Solution

It is desirable that the solution is able to operate at maximum performance over a wide range of periods and, in order to evolve a controller to do this, we must adapt the fitness function of the GA appropriately. It is thought that by developing a single controller that will work well in a series of different period monochromatic waves, we may produce a controller that will work well when excited by a more complex, polychromatic wave. To test this hypothesis we select four[5] monochromatic waves of periods, 3.5, 4, 5 & 7 seconds.

Each individual is evaluated in a 60 second simulation at each of these periods using a chunk size (see section 4.3.3) of 100ms. The total power developed in C_{PTO} for each period is then averaged to provide the fitness for each individual. The simulator is described in appendix A).

[4] $C_{PTOevolved}$ is determined by the genetic algorithm

[5] We used four separate waveforms as this provided a balance of a reasonable frequency range and reasonable computational intensity.

A real valued genetic algorithm was used (as described in section 4.4.1) to evolve the synaptic weights of the neural network with a population of 40 individuals and using basic genetic operators. It was found that when running this evolution on a single machine, each generation was taking up to 30 minutes to evaluate and, as we require between 500 and 1000 generations before we can observe a solution converging, this resulted in very long processing times. In order to speed up processing times, a multi-processing environment was adopted, as described in section 4.4.3 and allowed a 1000 generation simulation to be completed in approximately 24 hours.

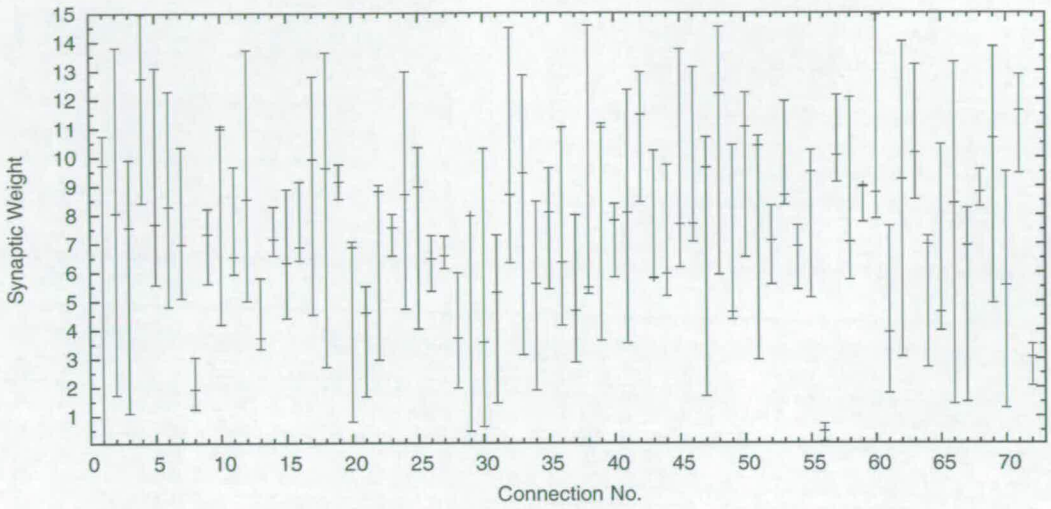
As has been described earlier, the genetic algorithm is a heuristic search technique based upon a random initial seed that does not guarantee an optimum solution, only a good one. Therefore in order to be confident in the results generated by such technique, and to be sure of avoiding false minima, three separate evolutions were invoked. Each of these converged within 500 generations and the average variation in the individual weights between these solutions can be seen in fig 5.2(a). The convergence of these solutions is shown in figure 5.2(b) and it can be seen that each of these controllers develop approximately the same maximum power *(within*$\approx$ 5%*)*. It is interesting to see that, even though the controllers generate almost the same ultimate performance, they each have a different synaptic structure, suggesting that the neural technique may be very versatile and illustrates the effectiveness of the GA to produce good solutions.

5.3 Further Latching Results

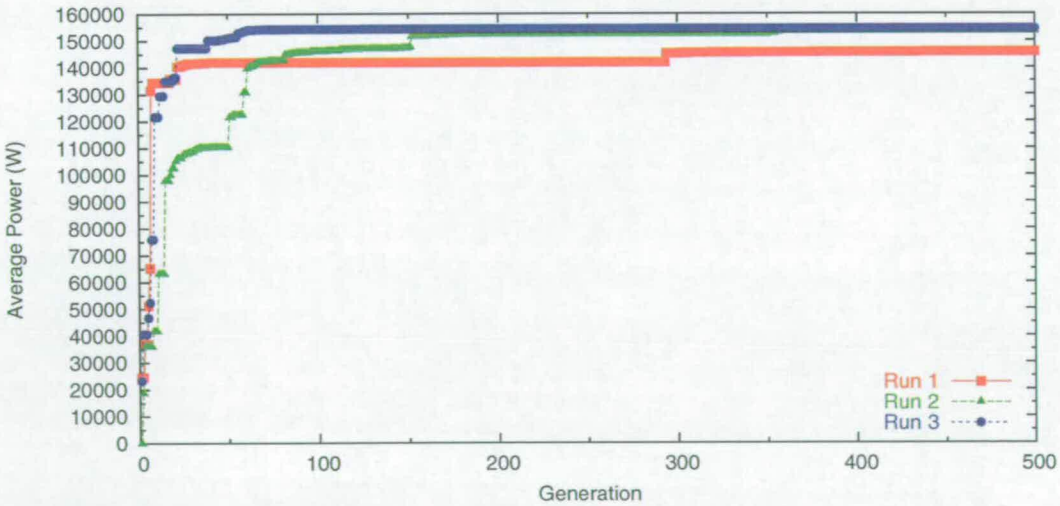
Figure 5.3 shows the best controller for each of the three evolutions and figure 5.2(a) compares the values produced for each synaptic weight. It can be seen that there is very little correlation between them, even though they each produce near-optimum results. Nevertheless, there are certain similarities between individuals, in that certain weights feature very little variation across the controllers. It is reasonable, therefore, to assume that these weights are in some way required for operation. These "significant" weights are illustrated in fig 5.3(a).

The evolved networks featured an almost fully connected structure and it seems logical to look at the strongest connections in order to compare their structures. These three controllers can be seen in figure 5.3(b) and show only the strongest synaptic interconnections[6]. As can be

[6]It is appreciated that the lower weight connections do have a significant effect, however we consider this simplification reasonable for the purposes of comparison.



(a) A comparison of the best individuals resulting from each of the three evolutions. Each bar represents the spread of values produced for each synapse.



(b) The progression of the GA showing the three separate evolutions of a 3.5s to 7s bandwidth controller.

Figure 5.2: *Summary of GA results for three 500 generation evolutions.*

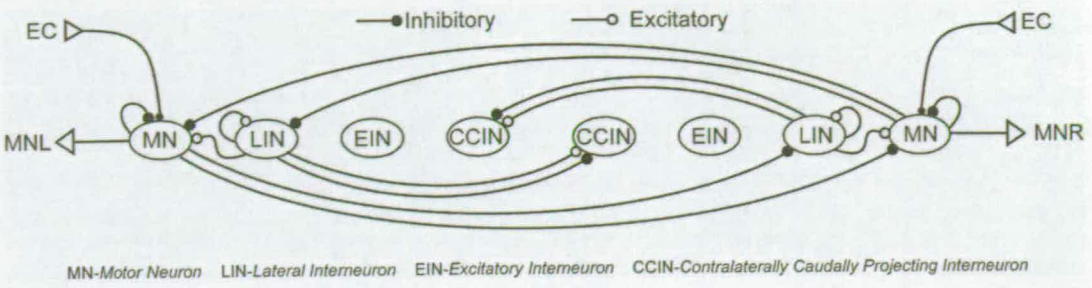
seen, this network bears little in common with the biological structure given by Grillner et al [46], except perhaps for the enforced symmetry. It is interesting, however, to note that in the biological controllers, the only contra-lateral connections were between the CCIN neurons, whereas here there is a proliferation of strong and long contra-lateral connections in all the networks.

These solutions were all evolved to produce viable latching control over a wide bandwidth. However, of these three controllers, #3 was chosen arbitrarily for further study. The latching performance in a 7s monochromatic wave is shown in fig 5.4, it can be seen to provide latch and release points in agreement with theoretical optimum latching and the same as observed with the "full-latching" controller shown previously.

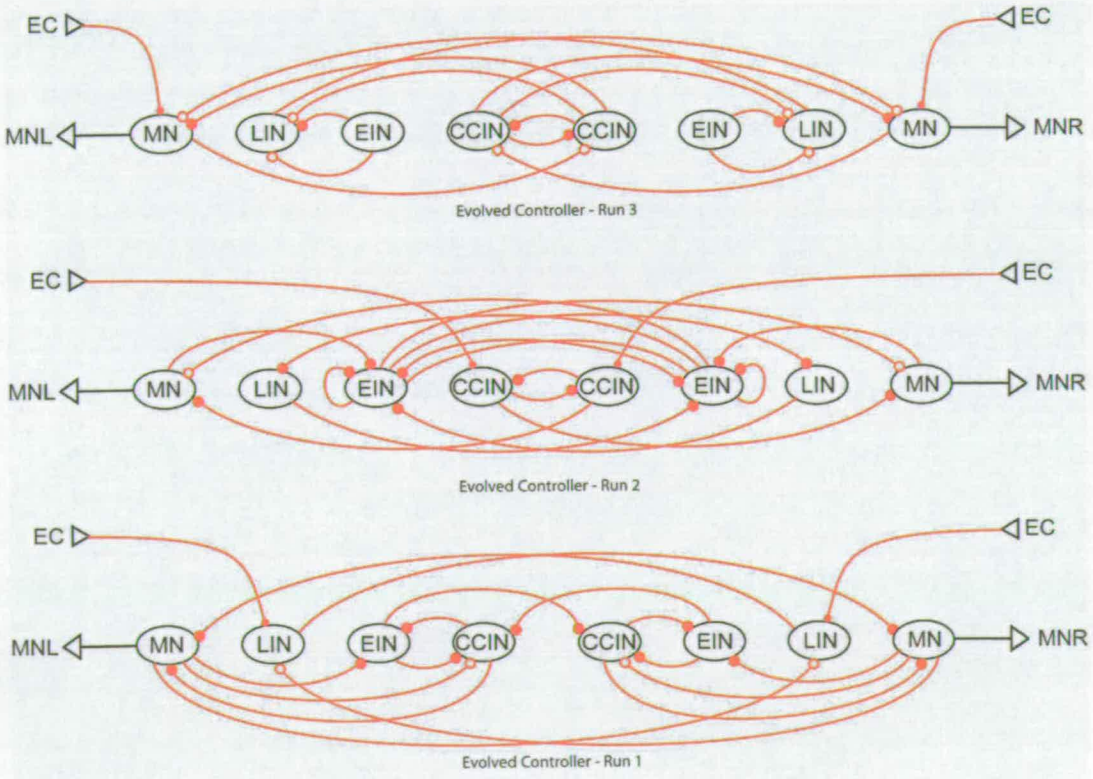
Figure 5.5 shows the performance of the evolved network compared to the natural undamped response of the mechanical system and also to the same mechanical system operating under optimum real damping. For the purposes of this thesis, optimum real damping is implemented as the maximum power obtained with a specific fixed damping level (see section 2.4). This plot illustrates the advantage gained by using a control method such as latching, illustrating very large performance gains shown throughout the bandwidth of the controller. It also demonstrates that the evolved neural controller is able to implement a latching system over a range of different period monochromatic waves.

It is important to note that this neural controller produces an appropriate latching strategy, without requiring any future knowledge of the wave. Whereas optimum real damping requires the wave period to be known before an optimum value can be determined. The phase of the system is shown in the upper plot and this clearly shows a near-ideal phase shift of $\approx 45°$ being maintained over the evolved bandwidth.

Although it is evident that the evolved controller can provide reliable performance in a fixed monochromatic wave, the next step is to evaluate the performance as the period of the input wave changes. In fig 5.6 we apply a waveform that sweeps across the frequency range of the controller. This swept wave is generated by stitching together a series of half-period waves of gradually increasing period that are joined together at each zero crossing. This is then decomposed into a discrete time series for the simulator. The swept wave starts at 3.5 seconds and gradually increases to 10 second period after 35s of simulation. The response to this wave is shown in figure 5.6 and clearly shows that the evolved controller is able to adapt to the changing



(a) The connections shown here exhibit minimal variation between evolved controllers. It is thought that these signify the connections that are required to ensure optimum performance.



(b) Structure of the three evolved controllers. Strong synaptic connections ($> 1.7 \times average\,weight$) are detailed. It is clear that they share no obvious common structure, however all produce very similar, near-optimum results.

Figure 5.3: *Network structures for evolved controllers with 3.5→7s bandwidth.*

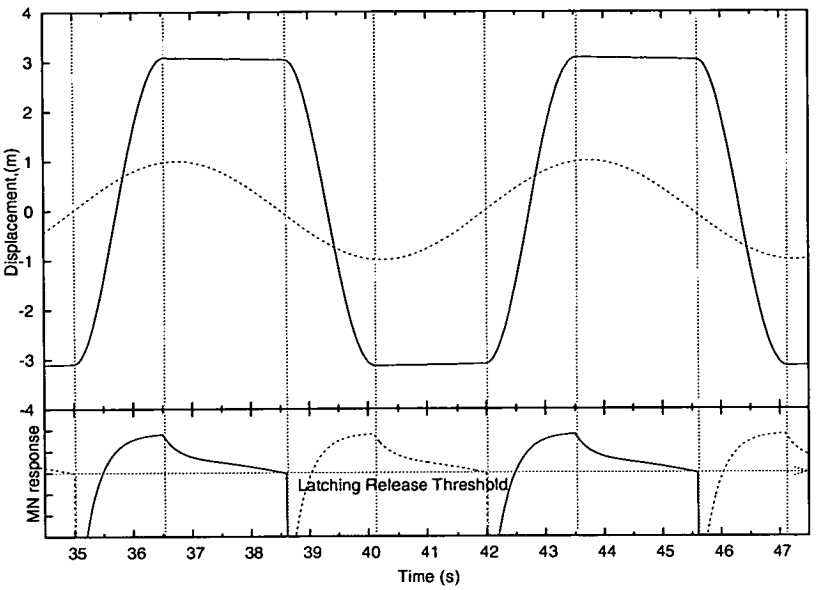


Figure 5.4: *Section of waveform with a 7s excitation showing buoy displacement (solid) with water elevation (dashed). In the lower plot the MN output can be seen with the latching transitions indicated by dotted vertical lines.*

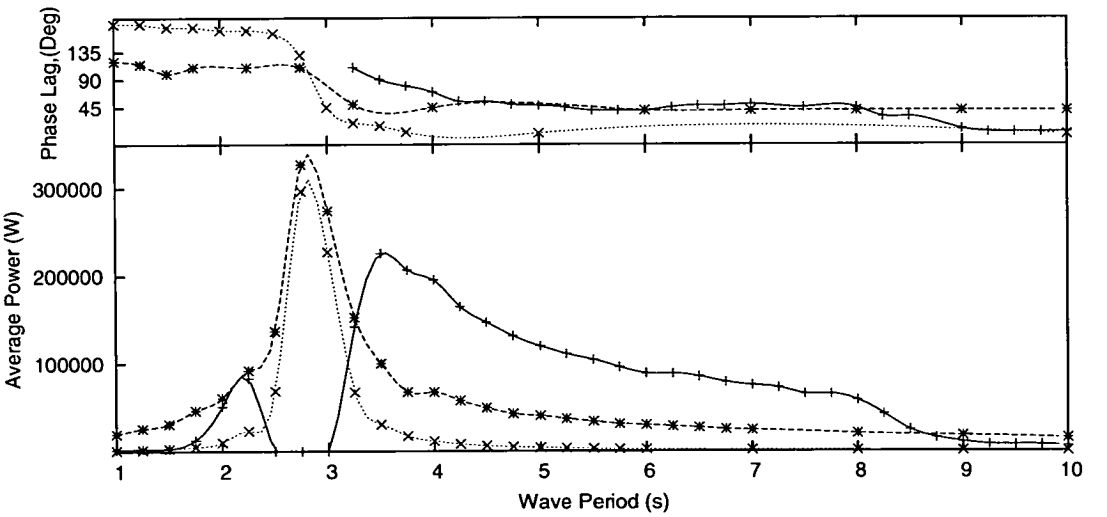


Figure 5.5: *Phase and power response for wave periods between 1 to 10 seconds. The neural controlled latching strategy (+), can be seen to outperform optimum real damping (∗) over much of this frequency range. The undamped system response is indicated by (×).*

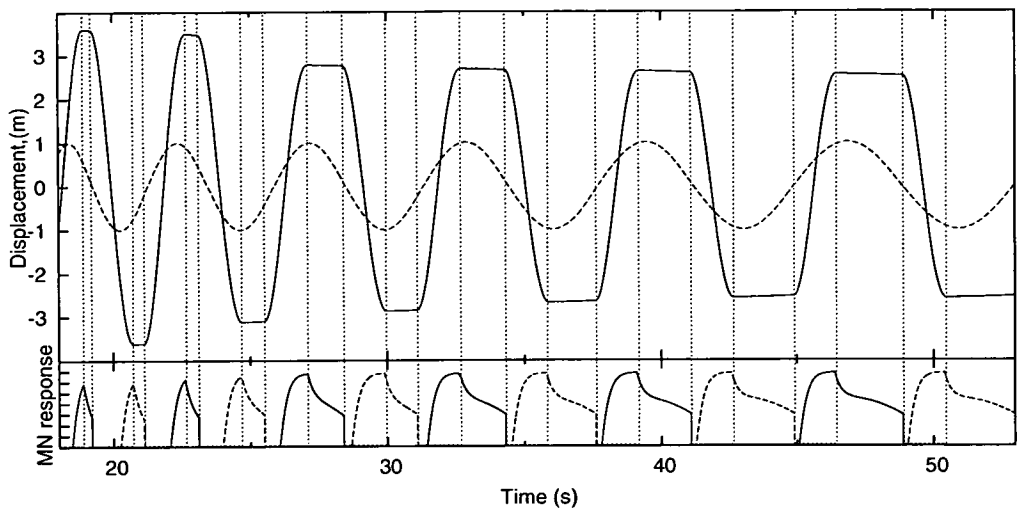


Figure 5.6: *Response of the system to a swept excitation over a 35s period. The wave period changes from 3.5 to 10 seconds over this duration. The top pane shows the wave displacement (dashed) relative to the device displacement (solid) while the lower pane shows the neural network output. The dotted vertical lines indicate the latch points for the system.*

input wave and maintain a latching strategy. Upon closer inspection it can be seen that the latching release occurs somewhat too early, appearing to use the previous cycle to determine the release point for the current one. This is to be expected and illustrates the feedback function within the controller. It shows that the neural controller uses the previous cycle to determine the correct activity for the current cycle, therefore it will be unable to produce an *optimum* latching response in a changing wave. Nevertheless this is a reasonable basis for a prediction and will allow the system to produce a noticeable performance increase in more realistic waves.

5.3.1 Performance Limits

The results in figure 5.5 show that the evolved strategy is successful over a useful range of frequencies, but it is interesting to note that it was not possible to produce a network that would provide latching control close to the resonant period of 2.8s. This can be explained relatively straightforwardly: if L_t defines the duration of latching and λ the wave period, then the phase lag introduced by latching can be described as: $\phi_i = \frac{L_t}{2 \cdot \lambda} \cdot \pi$. For maximum power transfer, the optimal phase difference between the wave force and the buoy displacement is $\phi = \phi_n + \phi_i =$

$\frac{\lambda}{4}$. At resonance, however, the natural phase shift of the buoy is $\phi_n = \frac{\lambda}{4}$, so the induced additional phase ϕ_i (though latching) should be 0 *at resonance*. Close to resonance, however, a lower limit is imposed upon L_t by the neuron model, forcing $\phi > \frac{\lambda}{4}$, resulting in sub-optimal power transfer. Under these conditions, since the latch-on time is too long, the difference between the buoy displacement and the water level is also too great. This effect introduces a feed-forward effect which reinforces this error - causing the latch-on time to get progressively longer which eventually results in the neural control becoming stuck in the latch-on position.

It is possible to adjust the neuron coefficients to reduce L_t but, due to the discrete nature of the simulation, data is only interchanged between mechanical and neuron models at the end of each major time-step (chunk), therefore L_t cannot equal zero as it is limited by the minimum major time-step (chunk-size) of the simulation.

As this effect is a function of the simulator, rather than the method it can clearly can be reduced by the introduction of smaller time-steps within the simulator. However, this will have the adverse effect of increasing the computational intensity of the model. Fortunately the periods of interest for the device are all longer than the resonant period. Therefore, this anomaly will have little effect upon the results[7]. The adjustment of neuron coefficients, in particular the time-constant, is investigated later and it can be seen that this does support the explanation given. We see that L_t is reduced and latching can take place closer to resonance without alteration of the time-step.

Also evident in the evolved controller was the reduction in magnitude of the motor neuron output towards the shorter periods. This occurs as the neural network is operated further away from its natural frequency of oscillation. The parameters for modelling each neuron, given in table 3.4.2, specifically the time constant, define this natural period of oscillation. This is due to the neuron not regaining the same charge as the delay between firing shortens. Conversely, the inverse occurs as the frequency decreases causing the MN output to reach a saturation. At the longest periods this effect causes the L&R MN pulses to overlap.

[7] A 'traditional' latching technique (see top diagram in fi gure 2.8) increases the phase delay between the device displacement and the wave. Therefore we select a device whose resonant period is shorter than the waves of interest so that such a 'traditional' strategy can be applied.

5.4 Improvements & Implementation

As we have seen, in order to alter the natural period and bandwidth of the neural network we can adjust the neuron time-constants and it can be seen that for this mechanical system, this has the effect of adjusting the bandwidth over which a latching solution can be found. The network used is based upon the biological model developed by Grillner [46], and the effects of scaling these neuron time-constants can be shown by linearly scaling each time-constant by a value, Nt_s. The neuron time-constant in effect only adjusts the natural frequency of the network and through modification of this scaling value it may be possible to develop controllers that will cover different periods.

Evolved Bandwidth		**Damping**	**Neuron Scaling Factor**
Start (s)	*Finish (s)*	C_{PTO}	Nt_s
3.1	6	8490	0.5
3.5	7	9431	0
6	10	14967	1.4

Table 5.2: *Variation in C_{PTO} against evolved bandwidths.*

Table 5.2 shows the scaling factors and associated parameters that are used to evolve new controllers in the same way as before. For each scaling value and bandwidth a controller was produced and the resulting frequency plots are presented in fig 5.7.

In figure 5.7, the different controllers can be seen to produce significant improvements within their particular frequency ranges. The smaller value of Nt_s has clearly enabled latching to take place closer to resonance, with the time-step issue (described in section 5.3.1) now being the only limiting factor. At the longer periods the performance is also clearly increased, although this improvement can be explained partly by the increased value for the power-takeoff damping C_{PTO}, rather than just the altered Nt_s. It is commonly known that as the period increases, the damping required to develop maximum power also increases. The changed neuron time-constant only makes effective latching *possible* at the longer periods and the increased value of C_{PTO} [8] allows the higher power to be developed.

Table 5.2 shows how the damping increases with the periods and this is further illustrated in figure 5.8 where the 3.1s-6s controller is shown with three differing values of C_{PTO}. This illustrates that for the longer periods, the value of damping must be increased, and can be done

[8] When compared to the other control shown and mainly due to the particular bandwidth used.

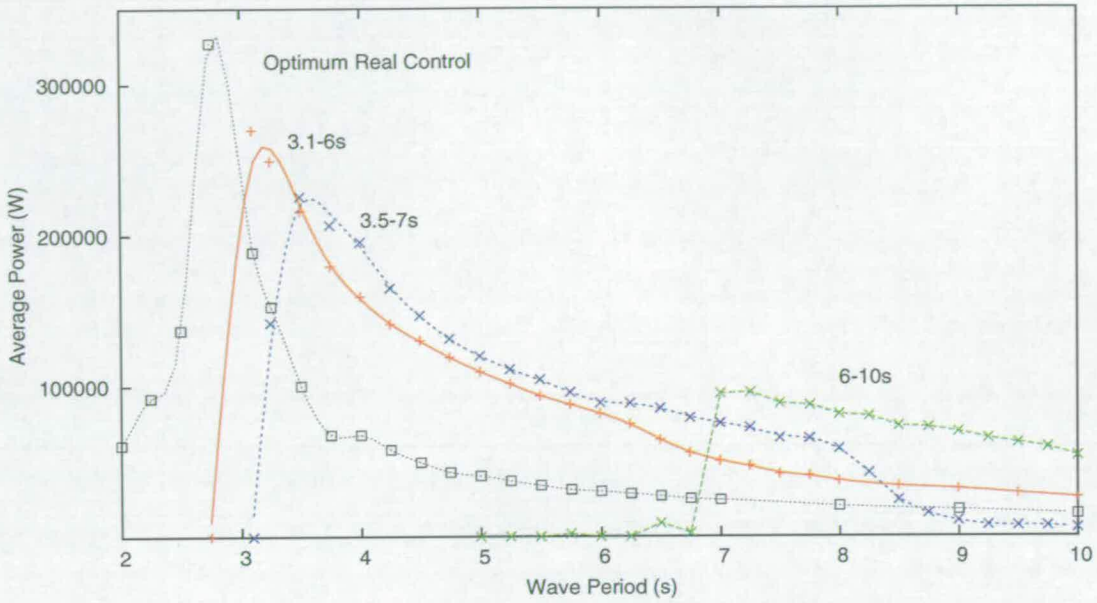


Figure 5.7: *Frequency response for evolved controllers using different neural scaling values.*

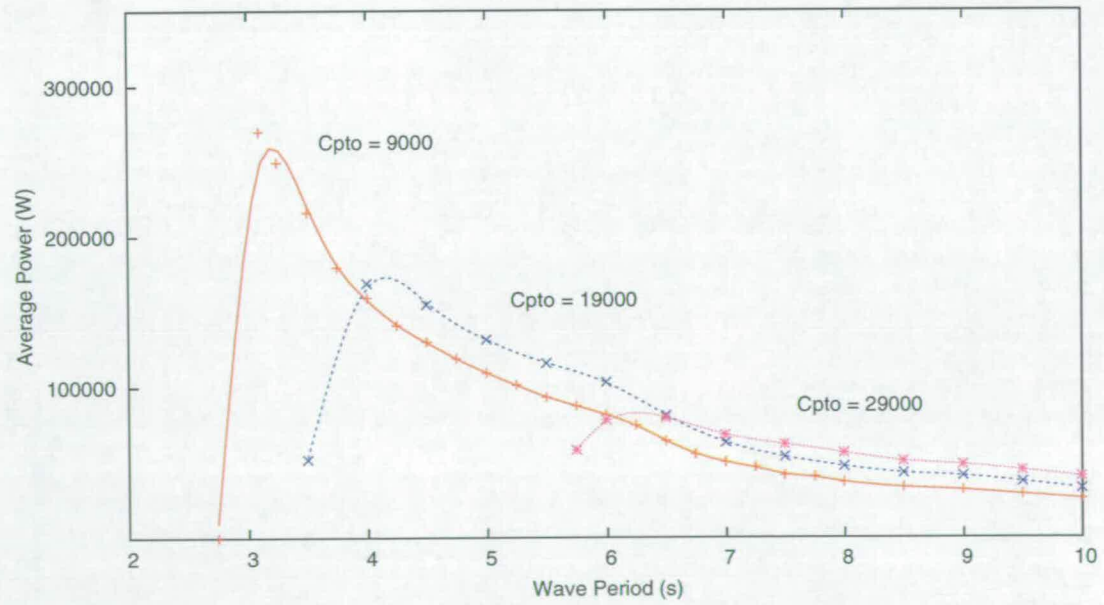


Figure 5.8: *Frequency response for controller evolved over a 3.1→6s bandwidth, using* $Nt_s = 0.5$*, illustrating the effect of varying levels of power take-off damping* C_{PTO}*.*

independently of the neural controller. It therefore becomes evident that the performance at longer periods is not limited solely by the neuron parameters and thus a higher C_{PTO} should be used.

This effect is is to be expected and is seen when optimal real damping is used. In this method a higher value of damping is required to maintain maximum power as the wave period moves away from the period of mechanical resonance. The resulting effect on this system means that, unsurprisingly, a single value of C_{PTO} will not be adequate for all wave periods.

5.5 Chapter Conclusion

This chapter describes initial work in using a neuronal network to implement a latching strategy for a simple point absorbing WEC. The results presented are particularly promising in simple monochromatic waves and show that a latching strategy can be effectively implemented to generate significant improvements over a tuned system (as opposed to controlled, see section 2.4) such as optimal real damping.

Starting by introducing the combination of the mechanical buoy model and the artificial neuronal network the text quickly moves on to discuss how evolutionary techniques were used to develop the synaptic weights to fit this application. The use of a GA in this respect was a good choice as solutions are ideally suited to be transcribed in string form. The initial tests used basic genetic operators, and although it was clear that this was an optimisation problem with a particularly large search space, the GA proved capable of finding good and applicable solutions.

Following the initial tests a more robust latching method was introduced along with the need to develop a controller that could work across a variety of wave periods. Again the GA was successfully used to evolve a solution that allowed for convincing latching performance across a range of periods. In order to confirm these results, a number of separate evolutions were completed which confirmed the maximum power to within 5%. Each of these separate evolutions produced a slightly different synaptic layout but with certain synapses that were consistent across all controllers and it was seen that all controllers operated visibly in the same manner.

The evolved controller demonstrated a wide frequency response with a certain idiosyncrasy close to the mechanical resonant frequency. However, it was shown that this was a function of the simulator and, although this could straightforwardly be avoided with more work, it was deemed that this effect is far from the periods of interest and can thus be neglected. The effects of scaling the neuron time-constant were also covered and it was seen that the value of $C_{PTOoptimum}$ contributed more to the performance at periods far from resonance than the neuron time-constant. Nevertheless, it was seen that appropriate neuron time-constants were required for each chosen bandwidth so that oscillation within the neural network would take place.

This same controller was also shown to produce good, although non-optimal results in a changing period wave. Evolved using a selection of different period monochromatic waves, it produced valid latching performance in a non-regular swept waveform, this is particularly promising and tests in more complicated irregular waves are covered next in chapter 6.

Chapter 6
Further Testing & Expansion to Multiple Degrees of Freedom

6.1 Introduction

Chapter 5 dealt with the preliminary development and monochromatic testing of a neural strategy to implement a latching strategy on a simple wave energy converter. This chapter expands this work though testing in irregular waveforms and expands the mechanical system to examine how neural control may be implemented in an articulated device.

Firstly, the irregular test waveforms are introduced and their production is explained. Then we examine the simple single degree of freedom system and how it performs in these conditions when compared to a simpler uncontrolled system. The mechanical model is then expanded to produce a simple analogy to a Cockerell raft, using two interconnected buoys. A controller is developed for this configuration and the interaction of this more complex model with regular and irregular waves is examined.

Finally we extend the mechanical model to produce an analogy to a simple articulated device with two joints. This is a more complicated arrangement using three interconnected buoys and we look at how a control strategy may be implemented for this model. We examine a number of methods to evolve a controller, including turning to more advanced genetic optimisation techniques before summarising these findings at the end of the chapter.

6.2 Polychromatic Response

6.2.1 Test Waveforms

It has been shown (in fig 5.6) that the neural controller developed in chapter 5 can be used to correctly implement a latching technique in varying regular waves. In order to to further analyse the performance of this controller we develop a series of three increasingly complex and realistic waveforms with which to test the system.

The first waveform is intended to be the most basic and thus it is a simple bichromatic waveform. Generated through the sum of two sines, such that $W_{out} = A_a sin(\omega_a t + \theta_a) + A_b sin(\omega_b t + \theta_b)$ where the subscripts a & b represent the two waves to combine. Strictly a regular waveform, we specify that both periods are within the evolved bandwidth of the controller, producing the waveform as shown in figure 6.1(b).

The second test waveform, shown in figure 6.1(d) is a more complex version of the first, being a sum of three sines. Given by $W_{out} = A_a sin(\omega_a t + \theta_a) + A_b sin(\omega_b t + \theta_b) + A_c sin(\omega_c t + \theta_c)$ with again the subscripts denoting the different waves to be combined. Again this is a regular waveform and its component frequencies are also taken to be within the bandwidth of the controller, it can be seen that the addition of an additional component produces a distinctly more complex waveform.

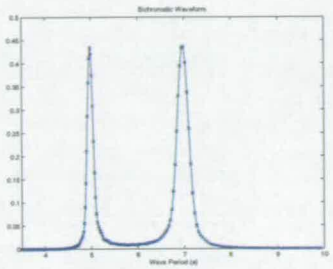
It is understood that neither of the previous waveforms are realistic. Therefore, for the final waveform we generate a time-series[1], from a Pierson-Moskowitch (PM) spectrum. This is an irregular spectrum created from a random seed and models a reasonably realistic, fully developed sea. The PM spectrum generated is shown in figure 6.1(e) showing a dominant period of 7 seconds. The time-series waveform derived from this spectrum is shown in figure 6.1(f).

These three test waveforms are all shown in figure 6.1 and are accompanied by their corresponding spectra which clearly show the component frequencies. We have seen previously (fig 5.6) that the neural controller can be used to correctly control a latching technique in varying regular waves. In order to study this method of control further, we observe performance in these three separate wave conditions and monitor the output.

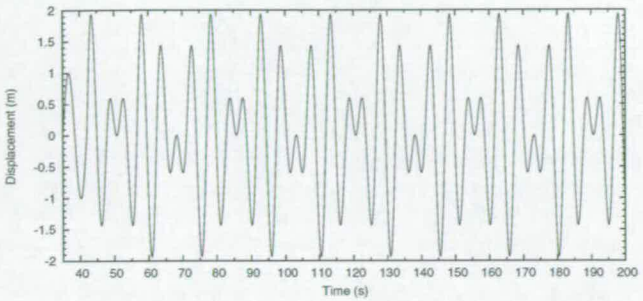
6.2.2 Bichromatic & Polychromatic Response

For each of the waveforms shown in figure 6.1, we evaluate the performance of the neural controller. For clarity we are using the same controller, evolved for a 3.5→7 second bandwidth, that was used for previous tests in the preceding chapter. We show the performance in each wave condition compared to the optimal real damping for each trace, which remains static for the whole simulation.

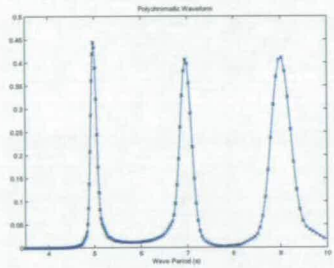
[1]This waveform was generated using the CS2 wave simulator [75] developed at Southampton university.



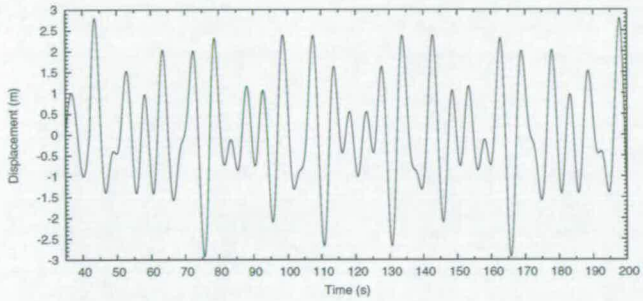
(a) Spectrum of bichromatic wave shown in (b)



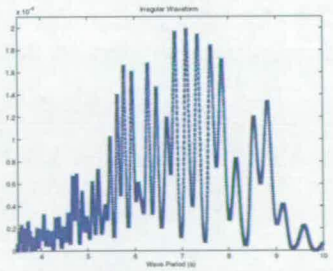
(b) Bichromatic wave combining waves of 5 and 7 second period



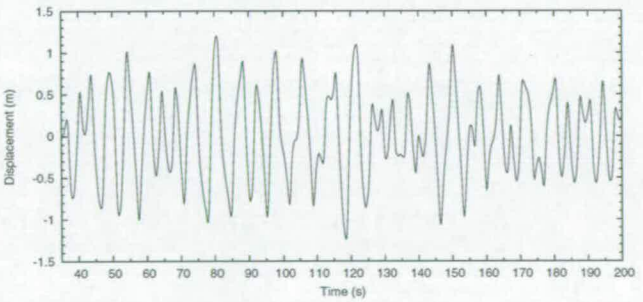
(c) Spectrum of polychromatic wave shown in (d)



(d) Polychromatic wave combining waves of 5, 7 and 9 second period

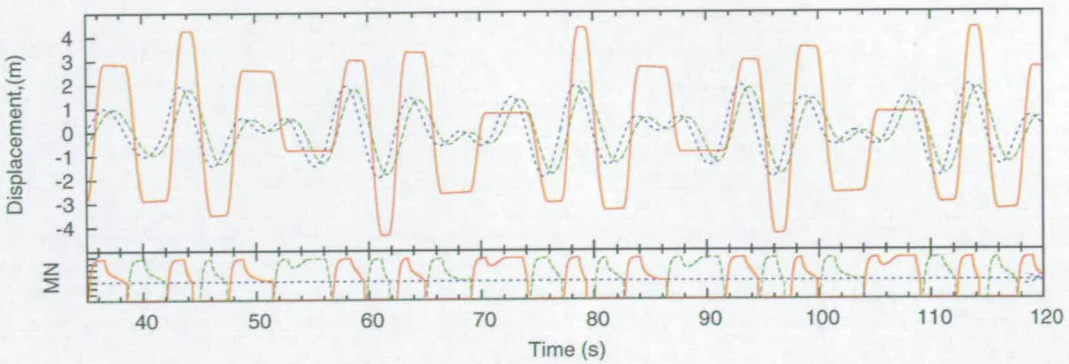


(e) P-M Spectrum of wave shown in (f)

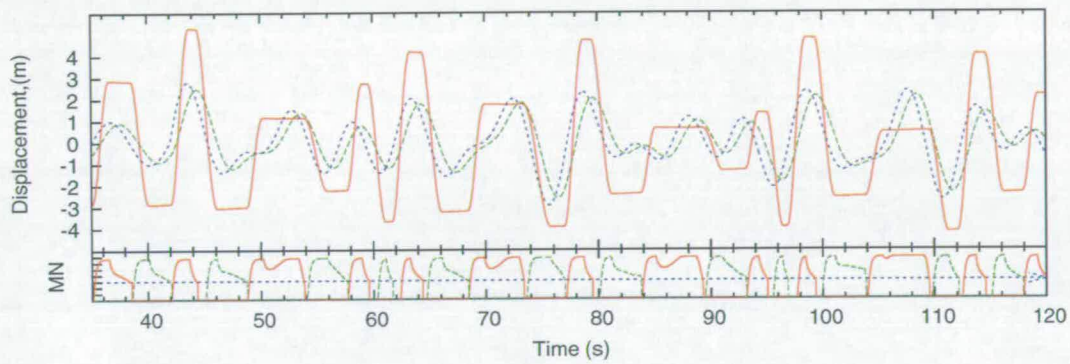


(f) Irregular Pierson-Moskowitch wave with a dominant period of 7s, produced using the CS2 wave simulator.

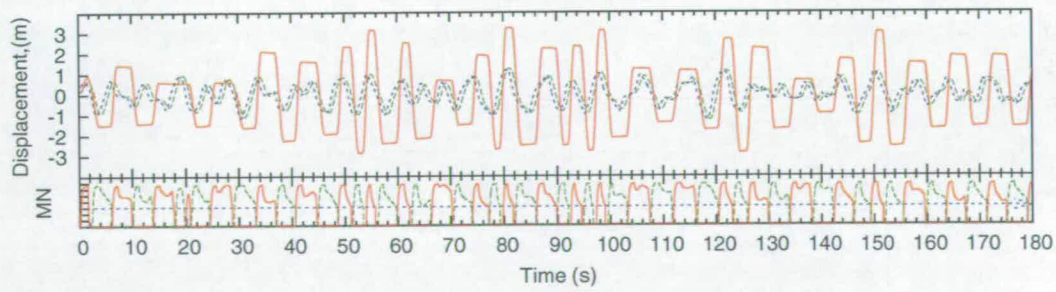
Figure 6.1: *Waveforms used for further testing.*



(a) Bichromatic wave: The power developed with optimal real damping averaged 50kW and with the controlled output it was 85kW



(b) Polychromatic wave: The power developed using this waveform increased from 60kW average with optimal real damping to 85kW with the neural control.



(c) Irregular wave. This is shown over a 180 second period and here the average for the controlled system was 43kW compared to the system under optimal real control, which produced an average of 17kW.

Figure 6.2: *Buoy displacement shown for the three different wave excitations. Each shown under optimal real damping (green) and under neural latching control (red), the surface displacement is given by the dashed blue line*

As can be seen in fig 6.2 the controller is able to adapt to non-regular waves with the neural control in general being able to find appropriate release points for the majority of wave conditions. It can be seen that large power gains are shown over optimal real control in all of the different waves, although the unrealistic polychromatic wave did not show as large improvement as the others. Automatic latch-on at zero velocity prevents excessive accelerations. However, this also means that the system is prone to latch-on at velocity inflections, rather than significant peaks. This effect can be seen at various points, most clearly in the unrealistic waveforms shown in figures 6.1(b) and 6.1(d). At these points this effect causes the control to *miss* the following significant peak as it is already latched however even with these mistakes the control is still able to adjust to the incident wave, although this is a point that can certainly be improved.

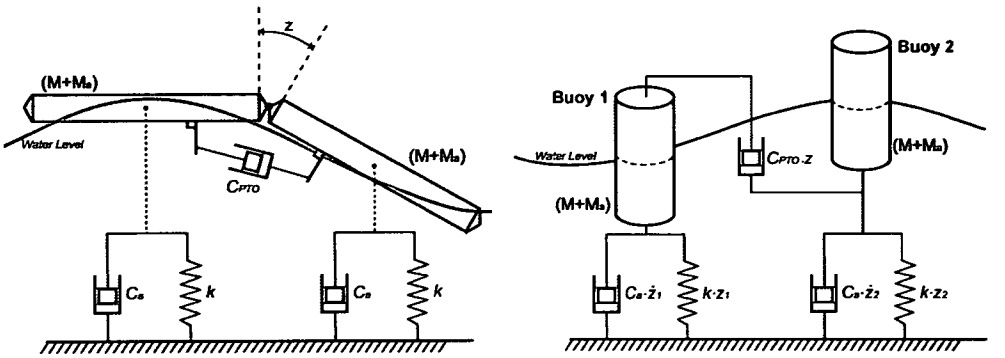
Most significant is the response the neural control shows in the realistic waves, illustrating an increase of greater than 100%[2] over the uncontrolled system (operating with optimal real damping). Although the performance in the bi- and polychromatic waves is interesting and illustrates certain flaws in the method, the fact that such a significant gain can be achieved in a realistic wave condition shows the potential of this kind of control.

6.3 Expanding into two buoys

So far we have considered a buoy operating in only a single degree of freedom and we have clearly demonstrated the potential of this method of control implementation, however our original inspiration came from the similarity between the wave induced motion of the Pelamis and the swimming motion of the Lamprey. In order to examine these similarities further we need to expand the mechanical simulation to more closely resemble a WEC consisting of articulated sections, our first step being as described in fig 6.3.

Here the relative angular movement between the two sections generates power in the damper C_{PTO}. This is a reasonably complex mechanical system. So, in order to provide a simple approximation, we can expand the current single buoy model by adding an additional buoy and using their relative motion to approximate the angular motion of the more complex system. Now the relative vertical velocity is used to develop power, resulting in a system comprising two single degree of freedom components which can be easily modelled, giving a system as shown in fig 6.3.

[2]Note: An increase of 100% relates to a doubling in magnitude



(a) WEC consisting of two connected rafts. The relative angular movement of the two, z, generates power in C_{PTO}.

(b) Two interconnected buoys providing an approximation to the connected body WEC shown in fi g 6.3

Figure 6.3: *Approximation to an articulated WEC design using two interconnected buoys.*

The equation of motion of the single buoy can be expanded and rewritten to take into account the new arrangement with subscripts denoting the individual buoys. As described above the two buoys are considered as completely separate bodies with the only common component the power take-off damper, C_{PTO} which is connected between them.

$$\ddot{z_1}(M + M_a) = Fb_1 - C_{PTO} \times (\dot{z_1} - \dot{z_2}) - (C_a \times \dot{z_1}) - k \times z_1 \quad (6.1)$$

$$\ddot{z_2}(M + M_a) = Fb_2 - C_{PTO} \times (\dot{z_2} - \dot{z_1}) - (C_a \times \dot{z_2}) - k \times z_2 \quad (6.2)$$

$$P_{(t)} = C_{PTO} \times (\dot{z_1} - \dot{z_2})^2 \times t \quad (6.3)$$

Although these two buoys are described as independent, their physical separation will determine just how large the relative movement will be. In an articulated device such as this, the buoys will be oriented one behind the other in a line perpendicular to the wavefront, the difference in displacement (and hence relative velocity) between the buoys will vary with the wavelength and will have a significant effect upon the power produced.

It can be seen that the wavelength (λ of a regular wave in deep water is related to its period (T) by the equation; $\lambda = \frac{T^2 \times g}{2\pi}$ [76] where g is the acceleration due to gravity. This allows us to calculate the appropriate relative displacement between the two buoys. It can be clearly seen

that the output of the system will be significantly increased when the wavelength is a multiple of the separation distance. For example, a physical separation of $\frac{\lambda}{2}$ will provide the maximum relative displacement between two bodies and hence the maximum power. This effect is more pronounced in a real articulated WEC such as OPD's Pelamis® [5]. This device consists of four connected sections, each 30 metres in length and it can be seen that it will experience performance "peaks" in wavelengths that are multiples of its section lengths i.e. 2λ and 4λ. This device is therefore able to provide a natural tuning to a variety of incident wavelengths allowing a good response at shorter ($\approx$ 60m) as well as longer ($\approx$ 120m) wavelengths providing a naturally wide bandwidth.

Physical Separation

As we have described, the separation between the buoys will have a distinct effect upon the power generated. Therefore, in order to decide upon this value we need to consider the estimated bandwidth of the system. We know that the resonant period of the buoys is just under 3 seconds and for the case of a single buoy we can produce effective latching control from 3 to 10 seconds. Therefore, if we take 3 seconds to be our minimum period, we can set the separation so that at this point buoy separation is equal to wavelength, knowing that at this period zero power will be produced. A 3 second wave period equates to approximately a 15m wavelength. Therefore this will be used as the separation between buoys. A few more periods and associated wavelengths are shown in table 6.1.

Period (s)	*Wavelength (m)*
3.5	19
4	25
5	39
6	57

Table 6.1: *Periods and wavelengths used for evolution of a solution using two buoys with a 15m separation*

6.3.1 Evolving a New Controller

As with the single buoy system, these two buoys still feature only one power take-off damper C_{PTO} for the purposes of latching and this can be considered the 'control element'. Latching control can be implemented in the same way as before and although the dynamics of the system

are very different we can in theory use the same method as before to evolve a controller for this system.

Of course, our first experiment involved applying the previous single buoy controller to this system and it was quite interesting to note that, although not optimised for this system, was able to produce latching control, although the power produced was barely better than uncontrolled optimal real damping. (see figure 6.5) It is perhaps understandable that this previous controller did not fare better as this two buoy system features certain differences, most obviously that the levels passed between the neural and mechanical systems will be significantly different to those observed with the single buoy system.

The neural and mechanical systems in the single buoy system are coupled by the buoy velocity ($\dot{z}$) being passed directly to the sensory input of the neural network, provided by the edge cell (EC). As the system is expanded to two buoys, power is now developed through the *relative* velocity of the two buoys, giving the velocity in C_{PTO} as $\dot{z_1} - \dot{z_2}$. Therefore when the buoys are moving in opposite directions, the maximum velocity will now be twice that of the single buoy.

As a result we can redefine the edge cell input to be the square root of the relative velocity.

$$EC_{input} = \sqrt{(\dot{z_1} - \dot{z_2})} \tag{6.4}$$

Adjusting the Fitness Function

When evolving a new controller, we must also take into account the effect of the buoy separation discussed earlier as there will be a significant peak in the power developed around 30m wavelength ($\lambda = separation \times 2$). This may preclude solutions being found for the controller at the extremes of the bandwidth and we must adjust the fitness function of the GA to take this into account. Currently the fitness of the GA is defined as the average power developed in a series of regular waves spread across the required bandwidth of the controller. This will promote individuals with exceptionally high power at specific frequencies and precludes others who offer a sub-optimal, but wide bandwidth.

As it is unlikely that a solution will be found that will be able to provide optimal power **and** a very wide bandwidth, then due to wide spectrum of ocean waves it is beneficial to promote

a wide bandwidth, even at the expense of a slightly sub-optimal performance. This will result in a controller that is more able to deal with variations in wave period[3]. To develop this, we introduce a transfer function that links the power developed to a fitness value for the GA with high bandwidth as the primary incentive, and maximum power as the secondary. In order to apply this, the power output from at each period undergoes the following transfer function shown in equation 6.5 and figure 6.4.

$$\text{Overall Individual Fitness} = \frac{\sum_{n=4}^{n=1} \sqrt{\frac{P_n}{\alpha}} \times \alpha}{4} \tag{6.5}$$

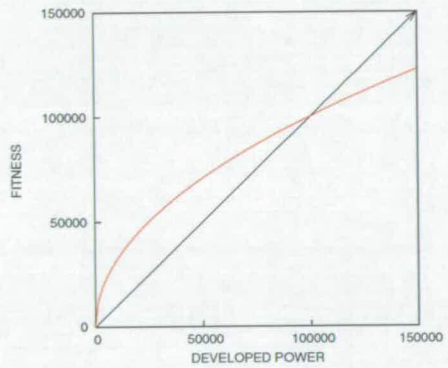


Figure 6.4: *This graph shows output power against fitness for a single individual, with scaling factor* $\alpha = 100k$

It can be seen that the transfer function will scale up power values below α and reduce those above. This provides extra incentive to develop controllers that will provide some response at all frequencies, and will not reward disproportionally those individuals that feature exceptionally high power at specific periods. It is worth noting at this point that in the same way as with the single buoy controller, if C_{PTO} is not modulated by latching, power is automatically reduced to zero. This excludes incorrect solutions that do not use latching control. The performance of this fitness function does depend upon the value of α and this can be defined as the point at which acceptable power has been achieved. This will of course vary depending upon the system and can be determined experimentally. For the system of two buoys here a value of 100,000 was chosen.

[3] In practice the expected range of wave periods will depend upon the intended location for any device.

6.3.2 Wide Bandwidth Controller

In a very similar manner to the single buoy controller we develop a wide bandwidth controller by choosing a series of frequencies spanning the desired frequency range which are shown with their accompanying wavelengths in figure 6.1. Using the same parameters as described in section 5.2.1.1, the GA converged to a maximum in a little over 600 generations, taking approximately 47 hours. The performance of the resulting controller is shown in figure 6.5. Here the average developed power, is plotted against period & wavelength. This is compared to optimal latching, to the response of the evolved single buoy controller developed previously and the output of the system using optimal real damping. The performance of the evolved controller is shown in more detail with a wave period of 5 seconds in figure 6.6.

The optimal latching response shown in figure 6.5 is the maximum power that can be achieved using a single tuned latching controller at each frequency. This is an estimate based upon the assumption that in a constant monochromatic wave[4] it is possible to develop a neural controller that will produce an optimal response. Accordingly for each discrete monochromatic period we can evolve a network that will produce the optimal latching response and available power. This power developed at each period can thus be approximated to theoretical maximum power, it is understood that this is an approximation and actual optimal response may be slightly higher but this is intended as a reference with which to compare and so can be considered adequate.

As expected, a significant peak in power is visible at wavelength of 30m due to the wavelength being double the separation. The controller does demonstrate a good response across the frequency range and although it is understandably less than the optimal response it is still significantly better than the system using optimal real damping. It is also interesting to see that the single buoy controller is able to produce a response across the range of periods however its response is only slightly better than that of optimal real damping.

The performance of the system in response to 5 second waves is shown in fig 6.6. In the upper plot the physical displacement of buoys (z) is shown, which clearly shows the effect of latching. The relative displacement is shown in the centre plot along with the water level at both buoys, this reacts as expected to the influence of latching, with the displacement being locked and released at the extremes of *relative* motion, providing approximately 90° phase shift from the

[4]Due to the fact that a constant monochromatic wave is 100% predictable and thus the wave itself provides all the future information required

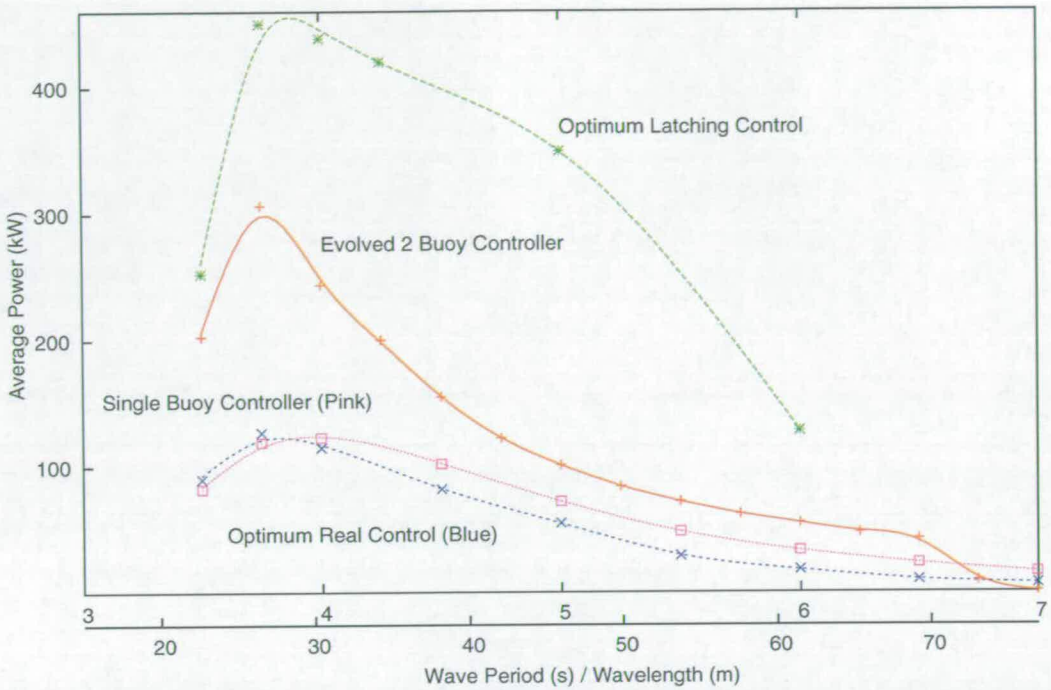


Figure 6.5: *Developed power against period and wavelength for the two buoy system using different controllers. The optimal output at each period for latching control in this configuration is shown in green.*

water level at the midpoint between the buoys. The bottom trace shows the input and output of the neural controller, showing the Edge Cell input relative to the MN output.

Real Wave Response

To complete the examination of this controller, we shall investigate its performance in a realistic, non-regular wave. As we have shown previously, a realistic waveform can be generated through a Pierson-Moskowitch spectrum that closely resembles a real wave. This spectrum can be expanded to illustrate the wave height at a single point on the surface of the ocean. As we need to develop a realistic wave for a system of two buoys with a down-wave separation, the wavelength as well as the period is now required. Ideally this requires two separate waves to be produced, one for each buoy, however as this spectrum has a clearly dominant period and wavelength we have simplified this waveform to use only the dominant wavelength for this test.

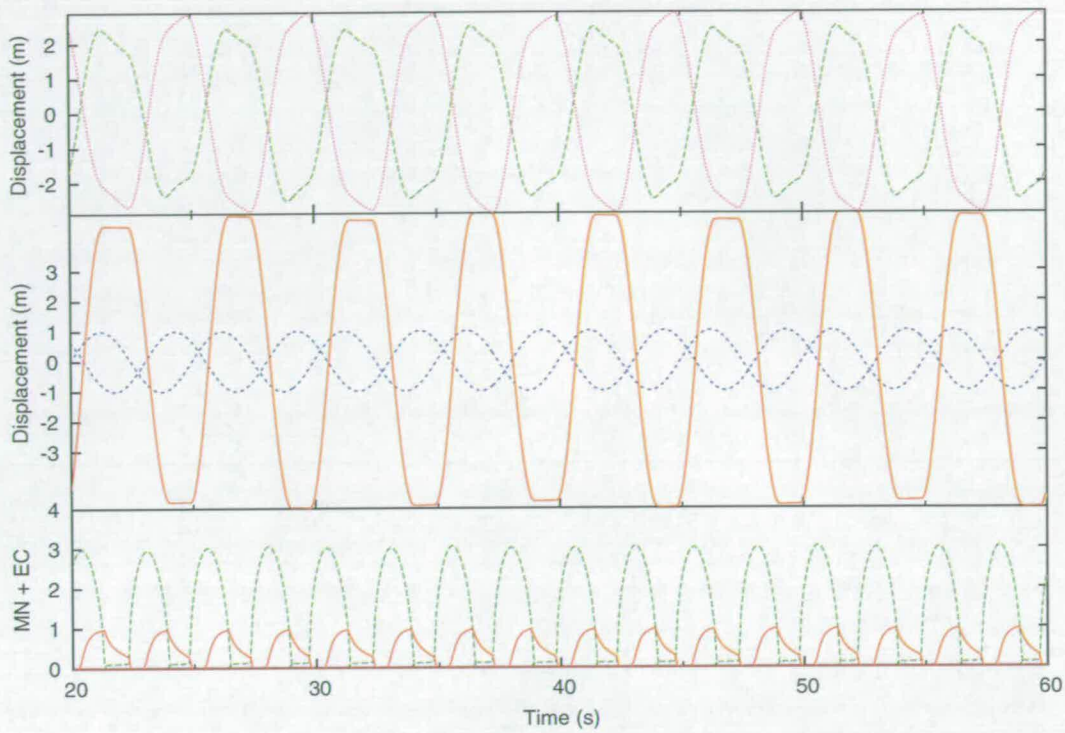


Figure 6.6: *Lower plot shows the EC input to the neural network (Green) alongside the MN output for a 5 second period wave . The middle plot shows the* relative *displacement of the buoys in red with the water level present at the midpoint between the buoys(blue), The upper plot illustrates the displacement experienced by each buoy, z_1(purple) and z_2(green)*

The performance of the controller in irregular waves is illustrated in figure 6.7[5], this illustrates a clear and quick adaption to the excitation waveform. The figure demonstrates a substantial power increase of 83% over optimal real damping. This is consistent with the high power increases seen when the single buoy controller is evaluated in non-regular waveforms as shown in figure 6.2.

[5]0-20 seconds are ignored as this is a confi guration period for the simulator and hence the output cannot be relied upon.

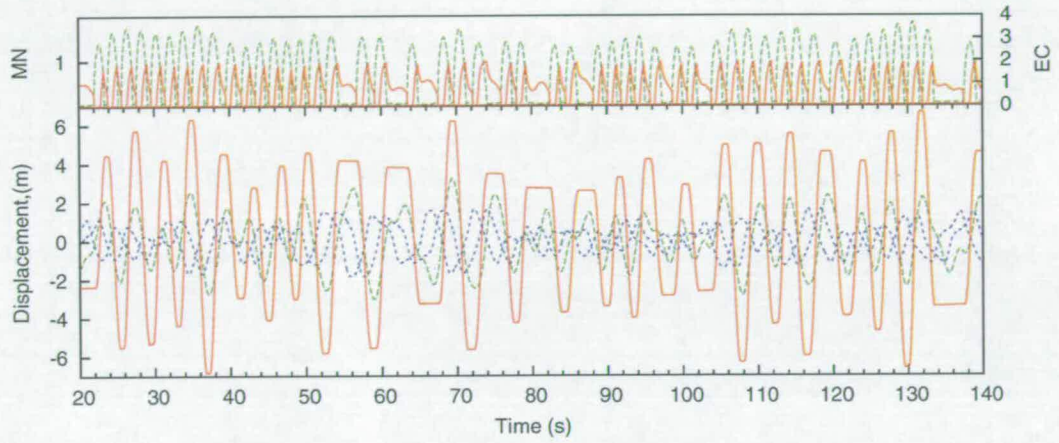
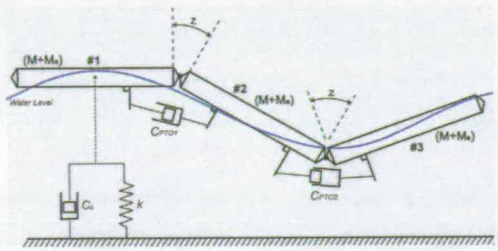


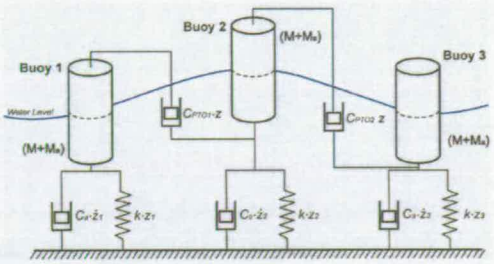
Figure 6.7: *The performance of the evolved controller in a 5.5s PM spectrum over 120s is illustrated here. The relative buoy displacement for optimal real damping is shown by the green trace (53kW average) and Neural latching control in red (98kW). The sea level at each buoy is given by the blue traces. The upper plot shows the controller EC input and MN output.*

6.4 Expanding to Three Buoys

So far the system of a single buoy and two interconnected buoys have been controlled by a single element and have illustrated how a biologically inspired neural controller can be used to implement a practical latching control strategy. This method, although tested in restricted conditions and with a simplified mechanical model, has shown significant improvements over real damping in irregular waves. The logical extension of the two buoy system is to encompass more interconnected units, as shown in figure 6.8(b). By extending the mechanical model in such a way we can start to roughly approximate the motion of real articulated WEC such as the PELAMIS by Ocean Power Delivery (see figure 6.8(c).).



(a) Mechanical arrangement of a three body articulated WEC.



(b) Simplification of a three body articulated system into three independent buoys.

(c) The PELAMIS WEC.

Figure 6.8: *Approximation to a three body articulated system.*

As the mechanical system is expanded to incorporate three buoys, it must also include an additional power take-off damper. This provides additional complication as we now have two control elements, C_{PTO1} and C_{PTO2}. The equations that describe the motion of the system are shown below.

$$\ddot{z_1}(M + M_a) = Fb_1 - C_{PTO1} \times (\dot{z_1} - \dot{z_2}) - (C_a \times \dot{z_1}) - k \times z_1 \quad (6.6)$$

$$\ddot{z_2}(M + M_a) = Fb_2 - C_{PTO1} \times (\dot{z_2} - \dot{z_1}) - C_{PTO2} \times (\dot{z_2} - \dot{z_3}) - \\ -(C_a \times \dot{z_2}) - k \times z_1 \quad (6.7)$$

$$\ddot{z_3}(M + M_a) = Fb_3 - C_{PTO2} \times (\dot{z_3} - \dot{z_2}) - (C_a \times \dot{z_3}) - k \times z_2 \quad (6.8)$$

$$P_{(t)} = C_{PTO1} \times (\dot{z_1} - \dot{z_2})^2 \times t + C_{PTO2} \times (\dot{z_2} - \dot{z_3})^2 \times t \quad (6.9)$$

The mutual dependence of motion is clear from these equations. It becomes obvious that if we modulate the value of C_{PTO1}, which controls the relative displacement of buoys 1 & 2, it will also effect the motion of buoy 3. The same obviously applies in that by modulating C_{PTO2} we affect the motion of buoy 1. This in effect produces a system of coupled oscillators that will provide a stable oscillation only given certain conditions. This is a complex system for which to find a stable control method and at first glance it seems to exclude the possibility that each power take-off damper may be independently controlled. Nevertheless, we can use the powerful optimisation techniques we have already employed to see if this is the case.

In this instance we must use a different method to control this model as there are now two *controlled* quantities, C_{PTO1} and C_{PTO2}, each of which will require a neural segment as a controller. Initially using a very simple configuration and assuming identical neural values in both controllers, we used a simple monochromatic excitation of 4s[6] which gives a wavelength of 25m.

For this initial test we use the GA as before to evolve the synaptic weights, however as we are now using two neural segments as the controller, in order to simplify the evolution we use the same synaptic weights in both[7]. As each controller will receive appropriate sensory input from the relative velocity of adjacent buoys, it is thought that this simplification is a reasonable starting point.

Figure 6.9 shows the output of the controller after a 1000 generation evolution, using the same GA technique as in earlier experiments. It can be seen that the controllers clearly attempt to latch the buoys in the same manner as in the single buoy controllers, however there is a clear

[6] 4s was chosen as this provided a good response for the single and two buoy models and is longer than ω_0.

[7] These two neural segments are also referred to as controller 1 and controller 2, which control dampers C_{PTO1} & C_{PTO2} respectively.

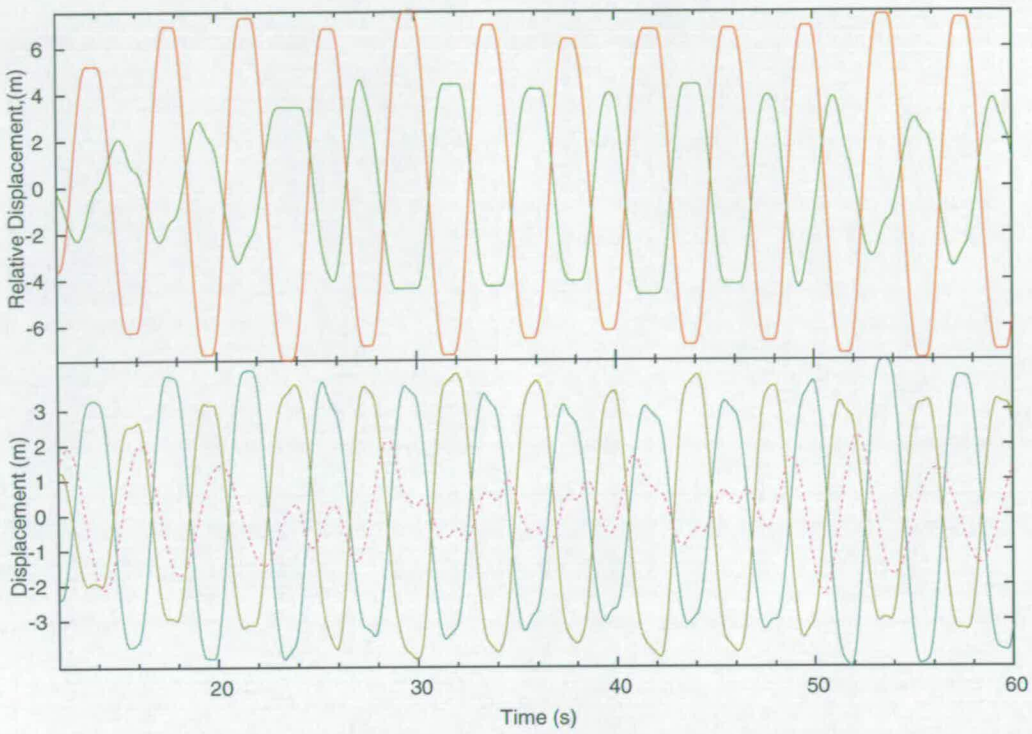


Figure 6.9: *Latching response of three connected buoys using two unconnected controllers(segments). The top trace shows the relative displacement of buoy 1&2 (red) and 2&3 (green). The lower plot shows the individual buoy displacements, buoys 1, 2 &3 are yellow, blue & dashed purple respectively.*

instability resulting from the coupled nature of the system which serves to disrupt the control. This results in an unpredictable and unstable output with the GA unable to find a stable solution for this configuration. Nevertheless the output power[8] did result in a average power of 419kW which was an increase of $\approx$ 110% over the optimal real damping configuration, however it was an improvement of only 10% over the two buoy system under neural control in the same monochromatic wave[9].

For further confirmation of this result, different synaptic weights were sought for both controllers, thus examining the possibility that this instability can be counteracted by using different controllers. This resulted in evolving double the number of synaptic weights and thus increased the search space significantly. Accordingly the GA took significantly longer

[8] In this confi guration almost all power was developed from the relative motions of buoys 1 & 2

[9] Due to the increase in buoys and the additional power take-off damper, a power increase of approximately 50% over the two buoy system should be expected

to converge to a solution. It was seen that the resulting controllers did illustrate differing structures but this did not result in a power increase over the previous result, developing a maximum of only 388kW. This supports the original suggestion that a three buoy system *requires* some form of coordinated control.

6.4.1 Biological Inspiration

As we have just shown, independent controllers are unable to produce an accurate latching strategy for a three buoy system. It may then be possible that some communication between controllers is required in order for them to adapt to such a complex system. If we look back at the biological inspiration behind this type of neural control we can see how it provides an analogue to this configuration (see section 3.5). The biological (and artificial) lamprey models feature individual segment oscillators that are connected together along the length of the body. These inter-segmental connections provide a phase delay between segments and ensure that a constant wavelength along the length of the lamprey body is maintained independent of the swimming speed. By allowing interconnections between the controllers in the same way as in the lamprey it is hypothesised that it will allow for stable control of three buoy system by allowing some form of information to pass between the individual C_{PTO1} & C_{PTO2} controllers.

In order to develop interconnections between these two controllers we must add an additional four matrices of weights, describing both inhibitory and excitatory inter-segmental connections and their direction. These matrices will be the same dimensions as currently used to describe the synapses within the controller. Figure 6.10 shows how these inter-segmental connections will work with a network of only 4 neurons.

With an individual controller size of eight neurons we can take advantage of the network symmetry as we have done so far and this will allow us to reduce each of these interconnection matrices to a 4×8 format rather than 8×8. Even so this provides a significant increase in variables (to be optimised)[10] from 74 to 202, vastly increasing the search space of the GA.

While developing his artificial lamprey model, Ijspeert [40] realised the problem of excessive search space and used a method to reduce the number of optimisation variables. He extended the weight of the intra-segmental connections caudally and rostrally (forward and backward) by an integer number of segments. Therefore, assuming that the connections are extended 1

[10]This still assumes that the two controllers will be identical as in previous artifi cial lamprey CPG models [6, 40]

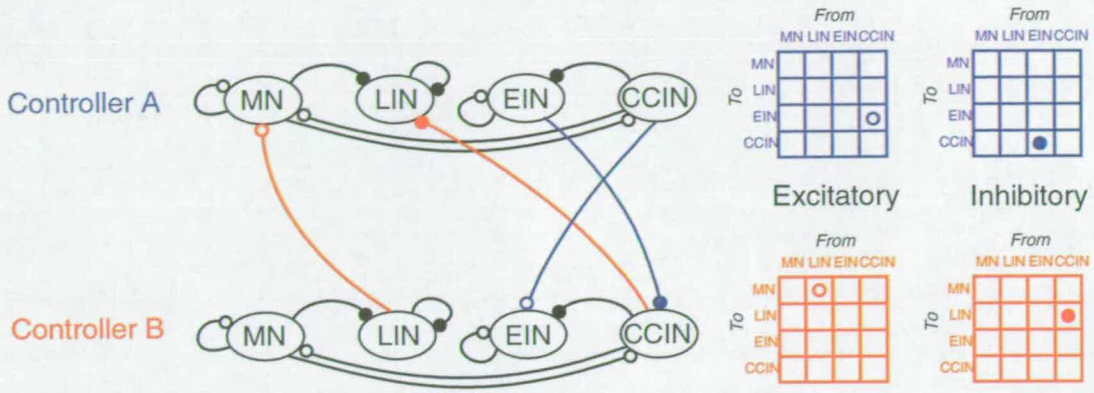


Figure 6.10: *Additional matrices required to represent synaptic connections between two controllers.*

segment; the weight of a synapse from LIN to CCIN in controller A is also applied to a synapse between neuron LIN in controller A and CCIN in controller B. Therefore an integer value can describe how many controllers a particular neural connection will extend to in a particular direction (caudal or rostral). In the simple system of two controllers a connection can only extend by a maximum of one segment, therefore each of the values in these inter-segmental matrices will be either connected (1) or not (0). As each matrix thus comprises one-bit binary rather than real-valued data, this reduces the complexity of the network significantly and keeps the increase in GA search space to a minimum.

In order to evolve a solution using this method of representation two methods are used: In the first method the segment weights are initially evolved as independent controllers (as developed above, see fig 6.9) then the inter-segmental connections are evolved separately. Following this the segment weights were then separately re-evolved using the best inter-segmental weights just developed. This procedure is repeated until no further improvements are seen. The second method simply involves the evolution of the intra-segmental and inter-segmental synaptic weights at the same time, thus being more straightforward but using a greater number of optimisation variables. Both of these methods use the same wave period of 4 seconds/25 metres and the same basic GA configuration as used in the previous 3 buoy experiments.

Result/Discussion

After convergence, the best individual from the iterative method generated 421kW average power and a response that was no more stable than with independent controllers. Initially

starting with the segmental weights evolved for the independent controllers, the first evolution of the inter-segmental weights produced the largest gain in power observed through any of these iterations. It was seen that the subsequent evolutions of the intra or inter-segmental weights produced very little improvement. After two iterations the GA was unable to make any improvement.

The 'all-in-one' evolution method failed to produce the same level of power, with the best individual developing only 388kW - It is thought that this is likely to be due to the increased search space allowing the GA to become stuck in a local maximum.

Relative to the 419kW produced through the independent controllers, this is an insubstantial increase and suggests that either Ijspeert's reduction technique does not have the finesse required to deal with control such as this, or the use of inter-segmental interconnections between controllers provides no benefit in this application. In order to eliminate the former theory, two additional methods are used to attempt to evolve a stable controller.

6.4.2 Further Investigation

As we have discussed in chapter 4, probably the most critical element of a Genetic Algorithm is the criteria by which it ranks individuals. This is the fitness function and up to this point this has been based purely upon average power developed in C_{PTO}[11]. It is thought that this may be promoting solutions that develop some latching strategy quickly and hence produce some power in one joint, rather than an even amount of power in both. This can be seen in figure 6.9 where latching between buoy 1 & 2 is successful, although between 2 & 3 it is unstable. This may be understandable given a coupled system such as this.

Optimising for Stability

An additional technique that is investigated involves changing the the fitness function to optimise stability rather than developed power. Although the main criteria for this application is power, we are particularly interested in a solution that will latch both joints in an accurate and stable manner, therefore an additional test is applied whereby the fitness criteria is adjusted accordingly. By rewarding individuals for developing an equal amount of power developed

[11] In the three buoy system fi tness is currently based upon the power developed in C_{PTO1} ***plus*** power developed in C_{PTO2}

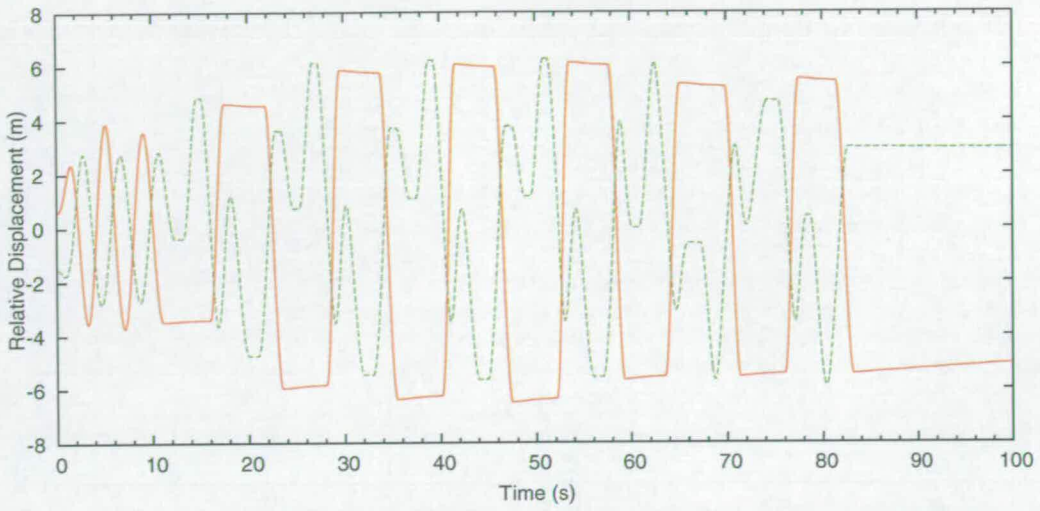


Figure 6.11: *Result of evolving for equal power in both joints. The synaptic weights were evolved over 60 seconds and beyond this period the graph shows a clear loss of stability. Joint 1 & 2 displacement is shown in red and green respectively*

in both joints (C_{PTO1} & C_{PTO2}) this may develop an overall stable latching control. This is applied simply by using only the power developed in the lowest output joint as the metric by which to compare individual solutions.

Using the additional technique described above to attempt to develop a more stable controller we produce the output shown in figure 6.11. Evolving over 1000 generations, this method used the same GA and configuration that produced the best result described above, however it used a different fitness function in order to try to promote stable[12] individuals. This function sacrifices developed power for stability and it can be seen that even given this compromise a stable solution was not produced. Latching does take place between the buoys and as we have observed in previous results, joint 1 (C_{PTO1}) is noticeably better than joint 2 (C_{PTO2}). The output however can be seen to be unstable, with the magnitudes of the buoys displacement varying even with a constant monochromatic excitation and then the output failing completely after 80 seconds.

This prompts further investigation as to whether there may be stable arrangement, therefore two further experiments are detailed, A & B:

[12] We define a stable solution as one where the power developed in both joints is approximately even and the latching response is constant and unchanging in a constant monochromatic wave (obviously excluding an initial startup period)

A

Assuming that Ijspeert's' method for implementing inter-segmental connections is not be suitable in this situation, a more direct method for developing the interconnections may be able to provide the required control. By developing the synaptic weights for the inter-segmental connections in the same way as the weights for the intra-segmental connections this will increase the number of variables to be optimised greatly, however it will be able to provide a much greater level of precision in the inter-segmental weights. This method involves the use of a real valued weight for each interconnection synapse and will result in an increase of 64 variables, significantly increasing the required search space for the GA and requiring a more advanced algorithm, such as a multi-population GA.

B

It may be that in order to develop a stable controller, the assumption that both segments should be identical is flawed. It may be that the controller may need to have two segments with slightly different characteristics in order to develop a stable system as well as having interconnections between segments. To do this, the two segments are no longer assumed to be the same and are thus evolved so they may develop different individual parameters. Firstly this will involve testing with Ijspeert's reduced interconnections method and then expanded to use real values independent interconnections as described in (A). These methods will result in an increasingly large number of synaptic interconnections to be developed, with the most complex resulting in a total of 270 real valued parameters being optimised by the GA.

In order to deal with these extremely large search spaces presented in these experiments, a number of techniques are employed, most importantly the use of a multi-population GA as described in (chapter 4). This uses a population of 80 individuals which are subdivided into four sub-populations, each of 20 individuals. The top 10% of fittest individuals are migrated between populations in a ring topology every 20 generations, therefore allowing an increase in genetic diversity and a more effective coverage of the increased search space. A non-redundant search modifier is also included that eliminates identical solution strings[13] to further increase diversity and reduce the possibility of becoming stuck in local maxima.

[13]In practice the non-redundant search was very effective early on in the search, within the first 200-300 generations, eliminating 1-2 duplicate solutions per generation, after this stage there were surprisingly few duplicates

Result/Discussion

Ultimately after several hundred hours of computer time no stable solution was produced using the majority of the methods outlined above. The best results from these methods produced consistently between 390kW and 420kW average power, however none of them could produce consistent latching at both joints, not even using a fitness function designed to produce even power in both dampers. The exception was found in method (B) when evolving both controllers and inter-segmental connections with separate real-valued weights. This was by far the most complex search conducted and converged to an average power of 455kW after 1500 generations and almost a week of processing on a farm of 20 linux workstations. Although this is an increase of only 10% over the unconnected, identical segments shown earlier it does produce a stable result which is shown in figure 6.12. Very interestingly, this result is a controller that in order to develop maximum power, removes the third buoy from the system by keeping the second joint unlatched and with zero damping. As the buoys are linked together only by the power-take-off dampers, this this effectively results in a two buoy system. It can be seen that the maximum power generated is in line with that produced with the two buoy simulation earlier[14].

Overall no stable solution was developed that can provide latching control in this articulated configuration, either with or without synaptic inter-segmental connections and even when sacrificing power to try to develop a more stable output. The best result we achieved featured an anomaly that causes the 2nd joint and 3rd buoy to be neglected completely - producing effectively a two buoy system, this output is shown in figure 6.12. We can confirm that the 3rd buoy has no effect of this system in this configuration as the power developed is the same as that produced in the two buoy results seen earlier (compare with optimal power in figure 6.5.). This is a surprising result and possibly indicates that this method of neural latching control is best suited to a single control element. The discrete nature of latching means that the neural EC input is zero for a significant part of the wave cycle and although a single segment may be able to effectively interpolate this missing information, two interconnected segments may not. This leads to the conclusion that it is not possible to produce a stable latching strategy for an articulated device with more than one control element using this method, and thus demonstrates a limitation of the latching technique.

[14] In the same wave, 4s 25m monochromatic wave, the two buoy simulation produced 440kW - see the optimal trace in fi gure 6.5

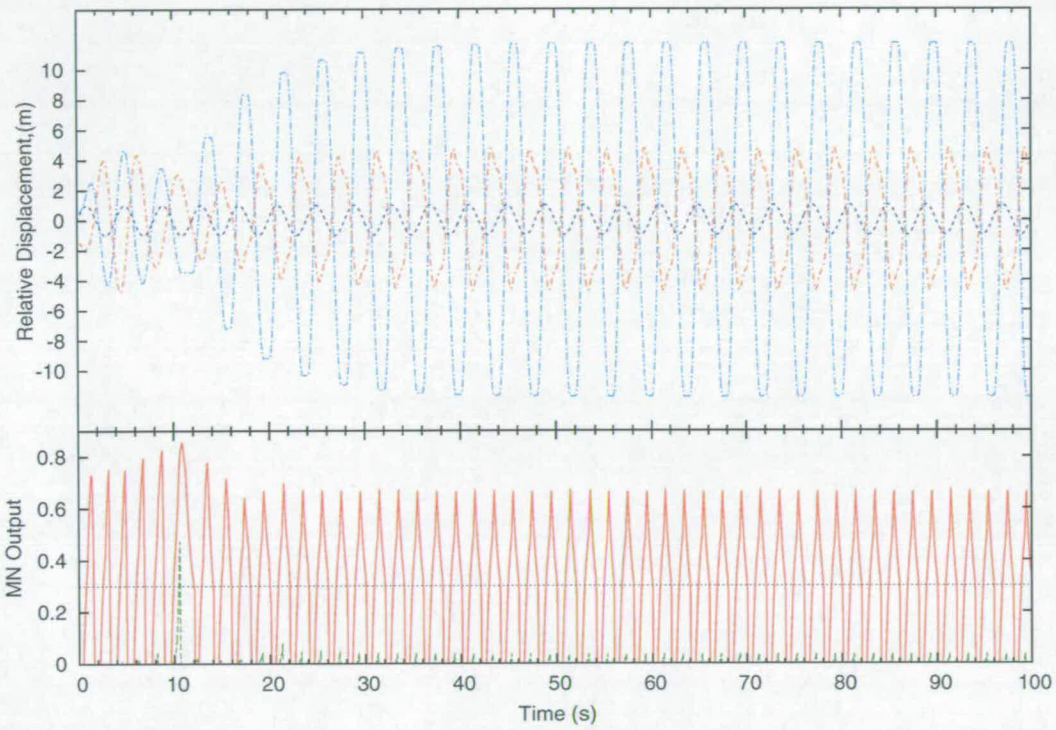


Figure 6.12: *Highest output result for three buoys. The top trace shows the relative displacement of the buoys, 1 & 2 light blue, 2 & 3 in orange. The MN response is at the bottom, note the spike on the green trace followed by lower-than-threshold output, this turns the latching off for buoy 2 & 3. The excitation wave is shown in blue*

6.5 Chapter Conclusions

This chapter has bought together the final results of this thesis and shows the performance of a single buoy in irregular waves right through to the expansion of the system to two and then three buoys in order to approximately model a multiple D.O.F. WEC, culminating by illustrating the limits of latching control in this application.

After highlighting a variety of increasingly complex waveforms, we started by examining the response of the previously evolved single buoy controller in these conditions. This illustrated the significant power increases that may be possible with this simple method. This also showed that the principle of neural control may be applied in a causal manner, where it does not require future knowledge of the input wave in order to calculate the release points for the strategy. It was seen that the system effectively bases its next estimate of release time based upon the previous cycle. This was adequate in all the waveforms tested to produce better power than an optimal real damped system and in the case of a Pierson-Moskowitch real sea produced greater than 100% increase over optimal real damping.

Following on from these results the single buoy model was expanded to use two buoys coupled by a power-take-off damper such that their operation was similar to the Cockerell raft (described in [29]). This system still only used a single power take out damper so it was a straightforward expansion of the previous implementation. It was demonstrated again that by evolving a controller that can produce power in a range of monochromatic wave periods a wide-bandwidth controller can be developed. It was also seen that the individual neuron parameters can be modified to adjust this bandwidth. The wide-bandwidth controller demonstrated good response in regular waves, although it was less than optimum (see fig 6.5), however as with the single buoy it provided good response in irregular waves.

The expansion of the mechanical model to encompass three buoys was the next logical step, although with mixed results. This model used two neural segments to control the two active dampers (C_{PTO1} & C_{PTO2}). After testing many different methods of evolution no significant and accurate latching response was seen, although power developed was still higher than produced with optimal real damping[15]. Ultimately the best response resulted in the evolved controller reverting to a two-buoy configuration and neglecting the third buoy. The hypothesis

[15]The maximum power developed with the three buoy model using the same waveform with optimal real damping was 178kW compared to the power of $\approx$ 400kW developed with the latching control.

that the interconnections between the two segments that made up the controller would help to pass information between the elements and thus produce a latching strategy in this complex configuration proved inconclusive. An increase of only 10% between unconnected and connected controllers was not large enough to conclusively prove this link.

Ultimately this chapter has increased the complexity of the mechanical model to the point where we can no longer find a latching control method that will produce a viable controller. However the performance of the single segment controller in a variety of waveforms has proved the validity of this form of neural control.

Chapter 7
Discussion & Conclusions

7.1 Discussion

This thesis set out to examine the plausibility of using a biological construct as the basis for a control mechanism for an articulated wave energy converter. In the process of doing so it has investigated further the application of latching as a control strategy and for a single buoy, has illustrated a methodology for evolving neural latching controllers that will perform well across a wide range of wave periods.

Through the work presented in this thesis, we succeeded in answering the following questions:

- Identify how the similarity between the lamprey neurophysiological structure and an articulated Wave Energy Converter may be exploited to provide a method of real-time control.
- Show whether the lamprey neural circuitry can be adapted to fit this control application through the use of evolutionary optimisation techniques.
- Investigate if the resulting controller can apply a *realistic* control mechanism to a simple mechanical WEC system.
- See if the resulting control system can be expanded and applied to an articulated WEC, similar to the Pelamis.

The following sections summarise the main findings of this thesis:

7.1.1 Control Analogy

The initial inspiration behind this work came from the observation that certain wave energy devices behave in a manner similar to the anguiliform locomotion exhibited in a number of creatures. In this instance the lamprey was taken as our subject as its biology has been

studied extensively and its neural structure has been modelled artificially. This previous work reveals a very simple neuronal structure which is divided into discrete segments that directly control the muscle output. The overall coordinated motion of the lamprey is the result of the interconnection between these segments which together form a neural construct known as a central pattern generator (CPG).

This primary analogy was formed by close inspection of the segmental nature of the lamprey CPG and relating it to the structure of an articulated wave energy converter. It was seen that the locomotion of the lamprey involved propagating a travelling wave along the length of its body, whereas the WEC responds to a sea wave travelling along the length of the device. Thus the initial similarities between the systems were formed.

In looking for a starting point to investigate this relationship further a much simpler system was studied. A single segment was taken and it was seen that by applying external sensory input, the segment was able to adjust and phase-lock its output frequency to match the frequency applied to the sensory input.

A simple but scalable model of a mechanical wave energy converter model was needed that would capture the essence of an articulated device, but would also provide a basic test-bed for initial concepts. For this, a single heaving buoy, tethered by a simple spring and damper approximation was used. It was found that the single segment neural network could be used in conjunction with this model by applying the model displacement to the sensory input and the period of the segment oscillation would become locked to the wave frequency.

It can be seen that the swimming speeds of the lamprey result in a much higher frequency than would be experienced with real waves. A lamprey CPG segment oscillates at 1→5Hz whereas we would expect wave frequencies to be in the region of 0.3→0.08Hz. In order to adjust the neural model to match the wave periods, it was found that the neuron time-constant could be altered to reduce the overall frequency. This was an arbitrary adjustment in order to provide a basic working model as soon as possible. Later work showed that further adjustment of the time-constant improved the performance of the system near resonance but also affected the system bandwidth. Further investigation into the neuron parameters could yield improved performance, however this was outside the reasonable scope of this thesis.

7.1.2 Synaptic Optimisation for WEC Application

The mechanical model used here is a simple heaving buoy approximation, which has been implemented to respond to any given waveform without requiring knowledge of the waveform in advance. In order to maintain a simple model this required a somewhat serious oversimplification of the hydrodynamic coefficients. This involved neglecting the frequency dependence of the added mass and radiation resistance and reducing them to static values. By modelling the buoy geometry in WAMIT the variation in added mass and radiation damping were calculated (see fig 2.11) and it was seen that for periods much longer than resonance, the variation in these hydrodynamic parameters was small. As the device had a resonant period much lower than the wave period of interest, and knowing that latching works best for wave periods longer than resonance, assuming that we do not consider the area around resonance (which in fact we cannot do due to a time-step issue with the simulator), then the variation in hydrodynamic parameters with frequency over the range of periods of interest will not have a *significant* effect upon the final results. It is known that by modulating the device with a strategy such as latching will have some additional effect upon these these parameters, however as this thesis is intended as a proof of concept, this is something that would be incorporated into a future, more complex model.

Once the mechanical and neural models were in place and the interconnection between the two systems was established, the method of control needed to be determined. As has been discussed, there are a number of control strategies other than latching control, such as reactive control that could have been implemented, however latching control was seen as the simplest to implement. In real devices latching can be used very effectively in single buoy application, or in configurations where there is only a single control element, however in articulated devices there is very little information about latching being effectively used. The final results emphasised the reasoning behind this and showed that in retrospect, a method of continuous control may have been a better choice in terms of scaling the strategy up to work with an articulated device.

The neural network's motor-neuron output was used to provide simple latch control to the damper, switching it between two values; when latched $C_{PTO} = large$ and when generating $C_{PTO} = optimum$. Using this rule the synaptic weights of the network were optimised by a genetic algorithm in order to determine the optimal configuration. The single segment comprises 144 synaptic weights but uses the inherent network symmetry to reduce this to to 72. A real-valued GA was used to evolve these weights, initially just using the average power

developed in C_{PTO} to rank the individuals. This gives a substantial search space of 10^{72}. We are unaware of an optimal structure for the CPG network, therefore a number of separate evolutions were completed each starting from a random seed, with the fittest individuals from each evolution compared to ensure convergence. It was found that ideal behaviour was possible with controllers that had significantly different structures, however it was seen that they all featured the same structure of contra-lateral inhibitory connections that is known to be a major mechanism in the oscillatory nature of the segment. For this case, as only one segment is used this variety of controller structures could be explained by the fact that there is no inter-segmental connectivity required so there will be a certain amount of synaptic redundancy in the network. The resulting structures and lack of inter-segmental connections also suggests that the sensory input, the Edge Cell, is the predominant burst terminating mechanism in this configuration.

7.1.3 Practical Control Application

It is clear that while developing performance in a monochromatic wave is interesting, is not adequate to enable the controller to perform in realistic multi-frequency waves. In order to investigate this a network was evolved using a series of monochromatic waves, each of differing period in the hope that the resulting controller will have the ability to adjust to polychromatic waves. For this evolution the fitness of the network was taken as the total power developed in all of these waves. It was found that as the shorter period waves generated a larger power in this system (purely as a result of the natural response of mechanical model) the result at each period had to be multiplied by a compensation factor to ensure the same weighting across the frequency range. This resulted in a network (controller) that demonstrated good performance in changing and irregular waves.

A latching strategy, although in practice quite straightforward, can provide significant difficulties in calculating the optimal power available due to its non-continuous nature so it was difficult to determine exactly how effective the neural controller was. Device excursions and induced phase were both in line with previous work, indicating that in a regular wave an optimal latching response could be produced. For convenience this assumes that the controller does produce an optimal response when evolved for a particular monochromatic wave, and this is used as the benchmark for which to compare the wide-bandwidth controllers against. It can be seen that it is quite reasonable to assume the neural controller can produce an optimal response in monochromatic waves, due to their nature. Due to the regularity each cycle will be

the same and thus the future wave elevation is known, resulting in a 'non-causal' system which is the requirement for any optimal control implementation.

This solution can be seen to be reliable as a series of controllers were evolved, each starting from a random population and all demonstrated the same ability[1] (within 5%) across the specified frequency range. When inspecting the performance of this evolved controller in a swept wave it clearly demonstrated a good latching response, with the power output substantially better than optimal real damping, however it could be seen that the technique produced a small non-optimal phase delay. This was also evident when the controller was presented with bi-chromatic & polychromatic waveforms, however the latching was evident and stable. In a reasonably realistic irregular waveform it was seen that even though the phase was sub-optimal, the resulting power was over 100%[2] better than the system under optimal real damping.

7.1.4 Expansion of Mechanical and Neural Models

Further work included modification of the mechanical model so as to more closely approximate an articulated device. This was implemented as a series of buoys, each modelled as previously but with the power take-off damper being connected between adjacent buoys, thus using the relative heave movement to generate power. Initially this work concentrates upon just two buoys as this arrangement only uses one control element (power take-off device). This two buoy system can therefore still use the controller developed previously with for a single buoy.

This new mechanical system adds two new components to the system. Firstly there is now a physical distance between the buoys so the wavelength as well as the period of the incident wave will affect the frequency response. Secondly, as a result of using relative movements, the power take-out device will be subject to a range of displacement that will be up to twice that of the single buoy.

Although it was reasonable to assume that the system in this configuration will still be very similar to that developed previously, it was logical to again evolve a wide-bandwidth controller and compare the results to the single buoy controller developed previously. Understandably the single buoy controller did not perform nearly as well as the controller evolved for this application, however it was still better than the optimal real damping for this system. These

[1] they each developed within 5% of the same power for a particular combination of input wave conditions
[2] Note: this thesis uses a 100% increase to indicate a doubling of magnitude

results were also compared to what we have considered to be the optimal performance at each frequency[3] and it was seen that the wide bandwidth evolved controller produced on average 50% of the ideal response in each monochromatic wave across its bandwidth. This response was clearly modulated by the effect of the changing wavelengths, with a significant peak as the physical separation between the buoys equalled $\frac{1}{2}$ wavelength which was to be expected as it is at this point that the relative displacement between the two buoys is a maximum.

The performance when tested in bichromatic and polychromatic irregular waves echoed the findings with a single buoy, showing large performance increases over optimal real damping. An irregular Pierson-Moskowitch generated wave illustrated an improvement of 85% over optimal real damping.

The next logical extension to add an additional buoy again added another complexity. Obviously the additional buoy increases the complexity of the equations of motion, but more significantly it adds an additional power take-off element that will need to be controlled. This extension now requires us to add an additional neural segment to control this second element. At this stage the original analogy is starting to emerge between an articulated WEC and the segmental nature of the lamprey and therefore it is now convenient to call the connections between buoys "Joints". With two joints the system dynamics are significantly different to those of the two buoy system so a new controller is required that includes an additional neural segment. A number of experiments were completed to test the limits and performance of this system.

This system is now quite complex as the act of altering one joint will now affect all three buoys, resulting in a critical system that that is difficult to control. As previously our first tests are carried out in regular waves in order to determine if the GA can converge to a stable, optimal solution. Initially it is attempted to develop a controller using two identical unconnected neural segments as the controller, which maybe unsurprisingly[4] produced an unstable output at joint 2, whereas joint 1 was a stable although poor latching technique. The next step involved

[3]These were developed by evolving a separate controller for each monochromatic period. As with a single buoy, these results can be considered to be optimum as they demonstrate the phase response that is expected from an ideal 'phase control' method such as latching. This is due primarily to the non-causal (regular) excitation and therefore the developed power and device excursions were also in-line with those reported as near ideal in other work. Considering the simplifi cations and approximations made in this model, this generalisation is assumed to be adequate.

[4]It is expected that the unconnected segments will fail to produce a valid strategy as the mutual dependence of the two control elements seems to require some information to be exchanged between the segments.

implementing interconnections between segments and this was done in three ways: Firstly by a simple technique developed by Ijspeert where the intra-segmental connection was extended (or not) to the adjacent segment. Secondly the inter-segmental weights are defined as a separate array of values and evolved alongside the intra-segmental neuron weights (which are identical in both segments).The third method is an extension of the second with the difference that it allows different weights in the two segments. The size of the solution string in the GA increases significantly as we go through these methods and as a result the GA needs to exploit more complex methods to enable a the search space to be effectively traversed. Ultimately, however, in all of these strategies no stable output in both joints was developed. Although a small increase 5-10% was observed in power developed when compared to the unconnected controllers, no symmetrical stable result was produced. This leads to an inconclusive verdict on whether or not the interconnections produced any benefit.

The third and most complex simulation involved the GA evolving every weight in the network (greater than 250 weights) and produced quite a surprising result. This showed that the fittest individual completely neglected the second joint. By not applying any latching to the second joint and setting the damping at zero resulted in power only being developed through the relative motions of buoys 1 and 2 and it was seen that the power developed was the optimum for a two buoy system. This interesting result seems to conclude that the only stable configuration for a latching strategy is involving one control element. This could be due to the discrete nature of the control in that by locking and releasing twice per cycle it is unable to develop a delicate enough arrangement within this reasonably unstable mechanical system.

Conversely it is also possible that the reason no stable result could be found is due to the how the latching strategy affects the feedback into the neural controller. As device movement provides the sensory feedback into the controller there will be no input while the joint is latched, ultimately producing a situation whereby once the joint/buoy is in this latched state the network must act upon past information in order to maintain the strategy. For a single neural segment with a single output, this appears to cause no problems, however it is possible that the added complication of the three interconnected mechanical elements might be causing the network to become unstable during the 'latched on' state.

7.1.5 Future Avenues

Although it has been shown that latching control is unsuitable for systems involving more than two buoys as described here, it is feasible that neural control may well be able to implement a method of continuous control. This type of control would provide a constant input into the network rather than the discrete values it receives via a latching method. It also seems likely that this may be able to provide the delicacy required to implement control in articulated devices as initially envisioned. The use of continuous control will involve a number of hurdles to be overcome, particularly how to relate the MN output into a device input that could be used to modify spring or inertia in a reactive control scheme.

Another area that could provide fruitful results is currently being investigated by Leena Patel[77]. This involves the modification of the neuron parameters in order to improve the response and performance of the segment. Although this work is currently concentrated upon the lamprey it is envisioned that it will be extended to further develop the controllers covered here.

7.2 Conclusion

To conclude, it is useful to reiterate the hypothesis that underpins the project: *Effective control implementation for Wave Energy Devices can be developed from (neuro)biological exemplars*

The conclusion of this work is that the hypothesis is true, with the following caveats:-

- Assuming a single segment of the lamprey CPG is used as the biological model.
- That we consider a single buoy system operating with a single degree of freedom that uses a single power take-off element.
- That the excitation wave is within reasonable limits set by the neuron time-constant and the mechanical system.
- That we define 'efficient' as developing a greater power than could be produced using an uncontrolled system.

The work presented in this thesis has delivered the goals outlined for it. It has determined that a neural network, derived from biology, can effectively implement a discrete control method for a single degree of freedom wave energy converter in irregular waves. This has been implemented in a highly simplified computer model, using an approximated mechanical model of a simple point absorber with simplified hydrodynamic parameters.

Furthermore it has demonstrated that evolutionary techniques may be effectively employed to optimise a neural controller for a WEC, demonstrating effective traversal of an extensive search space to produce very near optimal results in regular waves with convergence confirmed through repeated evolutions from a random population.

In addition, by illustrating the inability to produce reliable latching control in a system with more than one control element it has explored the limitations of discrete phase control. These effects are shown with direct reference to articulated devices consisting of two joints but with inference to the same effects when applied to devices consisting of more joints. This work demonstrates the inflexibility and impracticality of employing latching control to these type of devices.

The concept of neural control presented here highlights the plausibility of a ***realistic*** latching strategy that can be implemented in irregular waves in a causal manner. It is understood that the

resulting strategy is non-optimal, but when faced with trying to improve the cost effectiveness of WEC's in order to increase the viability of marine energy, this thesis introduces a simple and feasible method of significantly increasing the efficiency of devices[5] in a realistic environment.

This work also raises many questions related to the application of this neural control method, most importantly: Can the same principle of neural control be applied to a different control strategy? An investigation into the application of neural control on strategies other than latching could highlight more efficient and effective methods to control single and articulated wave energy converters. The effectiveness of the genetic optimisation techniques presented also raises the question: can a pre-evolved neural controller be re-evolved in an on-line (real-time) capacity? This could enable the a controller to adapt to changing sea conditions that were not present during an initial off-line evolution.

Ultimately this work has highlighted a route through which further study in this area can proceed. While it has shown that it is possible to take a simple biological construct and use it to apply latching control to a simple WEC, it has also highlighted that latching control *cannot* be used in more complex articulated devices. This method of 'neural control' has illustrated the ability to apply latching control in irregular waves, generating substantial power increases over current 'best practice'[6] methods.

[5]Assuming the device is using optimal real damping or another similar 'tuned' method of optimisation.
[6]Optimal real damping

References

[1] K. Budal and J. Falnes, "Wave power conversion by point absorbers," *Norwegian Maritime Research*, vol. 6, no. 4, pp. 2–11, 1978.

[2] M. J. French, "A generalised view of resonant energy transfer," *Journal Mechanical Engineering Science*, vol. 21, no. 4, pp. 299–300, 1979.

[3] R. E. Hoskin and N. K. Nichols, "Latching control of a point absorber," in *3rd International Symposium of Wave, Tidal, OTEC and small scale Hydro Energy*, (Brighton, UK), pp. 317–329, May 1986.

[4] A. Babarit, G. Duclos, and A. H. Clément, "Benefit of latching control for a heaving wave energy device in random sea," in *Proc. 13th Int Offshore and Polar Engineering Conf. ISOPE2003*, vol. 1, pp. 341–348, 2003.

[5] R. W. Yemm, R. M. Henderson, and C. A. E. Taylor, "The opd pelamis wec: Current status and onward programme," in *Proceedings of the fourth European Wave Energy Conference*, (Aalborg, Denmark), 2000.

[6] O. Ekeberg, "A combined neuronal and mechanical model of a fish swimming," *Biological Cybernetics*, no. 69, pp. 363–374, 1993.

[7] J. Ringwood and S. Butler, "Optimisation of a wave energy converter," in *Control Applications in Marine Systems*, (Ancona), pp. 155–160, IFAC, 2004.

[8] U. A. Korde, "Efficient primary energy conversion in irregular waves," in *Ocean Engineering*, vol. 26, pp. 625–651, 1999.

[9] L. Duckers, *Renewable Energy*, ch. 8, pp. 315–351. Oxford University Press, 2004.

[10] J. Falnes, *Ocean Waves and Oscillating Systems*. Cambridge University Press, 2002.

[11] J. Falnes, *Topics on Extraction of Ocean-Wave Energy*. Department of Physics, Norges Teknisk-naturvitenskapelige Universitet, 2002.

[12] J. Falnes, "Introductory lecture notes on ocean wave energy." Lecture Notes, June 2003. Short Course in Ocean Wave Energy, Gothenburg, Sweden.

[13] S. H. Salter, "World progress in wave energy - 1988," *International Journal of Ambient Energy*, vol. 10, no. 1, 1989.

[14] University of Strathclyde, "Small scale renewable energy systems for local energy supply." Internet: `http://www.esru.strath.ac.uk/EandE/Web_sites/00-01/small_scale_RE/`. Last Accessed: 19/04/06.

[15] W. J. Pierson and L. Moskowitz, "A proposed spectral form for fully developed wind seas based on the similarity theory of s. a. kitaigorodskii," *Journal of Geophysical Research*, vol. 69, pp. 5181–5190, dec 1964.

[16] K. Hasselmann, T. P. Barnett, E. Bouws, H. Carlson, D. E. Cartwright, K. Enke, J. A. Ewing, H. Gienapp, D. E. Hasselmann, P. Kruseman, A. Merrburg, P. Muller, D. J. Olbers, K. Richter, W. Sell, and H. Walden, "Measurements of wind-wave growth and swell decay during the joint north sea wave project (jonswap)," *Deutsche Hydrographische Zeitschrift*, vol. A12, p. 95, 1973.

[17] S. H. Salter, "World progress in wave energy - 1988," *International Journal of Ambient Energy*, vol. 10, no. 1, pp. 3–24, 1989.

[18] S. H. Salter, J. R. M. Taylor, and N. J. Caldwell, "Power conversion mechanisms for wave energy," *Proc. Institution of Mechanical Engineers*, vol. 216, pp. 1–27, 2002. Part M:J Engineering for the Maritime Environment.

[19] WAMIT inc. (MIT), "Tools for analysing wave interactions with offshore platforms and other structures or vessels." Internet: `http://www.wamit.com/`. Last Accessed: 19/04/06.

[20] D. V. Evans, "A theory for wave power absorption by oscillating bodies," *Journal of Fluid Mechanics*, vol. 77, no. 1, pp. 1–25, 1976.

[21] S. H. Salter, "Wave power," *Nature*, vol. 249, pp. 720–724, 1974.

[22] C. C. Mei, "Power extraction from water waves," *J. Ship. Res.*, vol. 20, 1976.

[23] J. N. Newman, "The interaction of stationary vessels with regular waves," in *Proc. 11 Symp. Naval Hydrodynamics*, (London), p. 491, Mech. Eng. Pub. Ltd, 1976.

[24] D. V. Evans, "Maximum wave-power absorption under motion constraints," *Applied Ocean Research*, vol. 3, no. 4, pp. 200–203, 1981.

[25] D. J. Pizer, "Maximum wave-power absorption of point absorbers under motion constraints," *Applied Ocean Research*, vol. 15, pp. 227–234, 1993.

[26] J. P. Kofoed, P. Frigaard, H. C. S. rensen, and E. Friis-Madsen, "Development of the wave energy converter - wave dragon," in *International Offshore and Polar Engineering Conference*, 2000.

[27] R. C. T. Rainey, "The pelamis wave energy converter: It may be jolly good in practice, but will it work in theory?," tech. rep., WS Atkins Oil & Gas, c.2000.

[28] Ocean Power Delivery, "The pelamis wave energy converter." Internet: `http://www.oceanpd.com/Pelamis/default.html`. Last Accessed: 19/04/06.

[29] M. Wooley and J. Platts, "Energy on the crest of a wave," *New Scientist*, pp. 241–243, 1975.

[30] J. N. Newman, "Absorption of wave energy by elongated bodies," *Applied Ocean Research*, vol. 1, no. 4, pp. 189–196, 1979.

[31] P. Haren and C. C. Mei, "Wave power extraction by a train of rafts: hydrodynamic theory and optimum design," *Applied Ocean Research*, vol. 1, no. 3, pp. 147–155, 1979.

[32] U. A. Korde, "Active control in wave energy conversion," *Sea Technology*, pp. 47–52, 2002.

[33] J. Falnes, "Optimum control of oscillation of wave energy converters," in *11th International Offshore and Polar Engineering Conference*, June 2001.

[34] P. Nebel, "Maximising the efficiency of wave energy plant using complex conjugate control," *Proc. Instn. Mech. Engrs, Part I, Journal of Systems and Control Engineering*, 1992.

[35] S. Naito and S. Nakamura, "Wave energy absorption in irregular waves by feed-forward control system," in *IUTAM Symposium "Hydrodynamics of Ocean Wave Energy Utilisation"* (D. V. Evans and A. F. de O Falcao, eds.), pp. 269–280, Springer-Verlag, 1985.

[36] U. A. Korde, "On the control of wave energy devices in multi-frequency waves," *Applied Ocean Research*, pp. 132–144, 1991.

[37] D. A. Guenther, D. Jones, and D. G. Brown, "An investigative study of a wave energy device," *Energy*, vol. 4, pp. 299–300, 1979.

[38] R. E. Hoskin and N. K. Nichols, "Optimal strategies for phase control of wave energy devices," in *Proc. Int. Symp. Utilisation of Ocean Waves-Wave to Energy Conversion*, pp. 184–199, 1987.

[39] T. R. Mundon, A. F. Murray, J. Hallam, and L. N. Patel, "Causal neural control of latching ocean wave point absorber," in *Proceedings of the International Conference of Artificial Neural Networks*, LNCS 3697, (Warsaw), pp. 423–429, September 2005.

[40] A. J. Ijspeert, *Design of artificial neural oscillatory circuits for the control of lamprey- and salamander-like locomotion using evolutionary algorithms*. PhD thesis, University of Edinburgh, 1998.

[41] S. Grillner, "Neural networks for vertebrate locomotion," *Scientific American*, January 1996.

[42] J. H. Or, *An Investigation of artificially-evolved robust and efficient connectionist swimming controllers for a simulated lamprey.* PhD thesis, The University of Edinburgh, 2002.

[43] F. Delcomyn, "Neural basis of rhythmic behaviour in animals," *Science*, vol. 210, pp. 492–498, 1980.

[44] C. M. Rovainen, "Neurobiology of lampreys," *Physiological Reviews*, no. 59, pp. 1007–1077, 1979.

[45] S. Grillner, "Neurobiological bases of rhythmic motor acts in vertebrates," *Science*, vol. 228, no. 4696, pp. 143–149, 1985.

[46] S. Grillner, T. Deliagina, O. Ekeberg, A. E. Maneira, R. H. Hill, A. Lansner, G. N. Orlovsky, and P. Wallen, "Neuronal networks that co-ordinate locomotion and body orientation in lamprey," *Trends in Neuroscience*, vol. 18, pp. 270–279, 1995.

[47] S. Grillner, P. Wallen, and L. Brodin, "Neuronal network generating locomotor behaviour in lamprey: Circuitry, transmitters, membrane properties, and simulation," *Annual Review of Neuroscience*, no. 14, pp. 169–199, 1991.

[48] J. T. Buchanan and S. Grillner, "Newly identified 'glutamate interneurons' and their role in locomotion in the lamprey spinal cord," *Science*, pp. 312–314, 1987.

[49] P. Wallen, "On the spinal generation of locomotion in fish," in *Neurophysiological Mechanisms of Locomotion*, 27th International Congress on Physiology, (Paris), 1977. Satellite Symposium.

[50] P. Wallén, O. Ekeberg, A. Lansner, L. Brodin, H. Travén, and S. Grillner, "A computer-based model for realistic simulations of neural networks. ii. the segmental network generating locomotor rhythmicity in the lamprey," *J. of Neurophysiology*, vol. 68, no. 6, pp. 1939–1949, 1992.

[51] T. L. Williams, "Phase coupling by synaptic spread in chains of coupled neuronal oscillators," *Science*, vol. 258, pp. 662–665, october 1992.

[52] A. H. Cohen, G. B. Ermentrout, T. Kiemel, N. Kopell, K. A. Sigvardt, and T. L. Williams, "Modelling of intersegmental coordination in the lamprey central pattern generator for locomotion," *TINS*, vol. 15, no. 11, pp. 434–438, 1992.

[53] S. Grillner, A. McClellan, and C. Perret, "Entrainment of the spinal pattern generators for swimming by mechano-sensitive elements in the lamprey spinal cord in vitro," *Brain Research*, vol. 217, no. 2, pp. 380–386, 1981.

[54] O. Ekeberg, P. Wallen, A. Lansner, and H. Traven, "A computer based model for realistic simulations of neural networks," *Biological Cybernetics*, no. 65, pp. 81–90, 1991.

[55] J. T. Buchanan, "Neural network simulations of coupled locomotor oscillators in the lamprey spinal cord," *Biological Cybernetics*, no. 66, pp. 367–374, 1992.

[56] N. Kopell, G. B. Ermentrout, and T. L. Williams, "On chains of oscillators forced at one end," *SIAM J. of Applied Mathematics*, vol. 51, no. 5, pp. 1397–1417, 1991.

[57] R. Jung, T. Keimel, and A. H. Cohen, "Dynamic behaviour of a neural network model of locomotor control in the lamprey," *J. of Neurophysiology*, vol. 75, no. 3, pp. 1074–1085, 1996.

[58] B. W. Kernighan and D. M. Ritchie, *The C Programming Language, Second Edition.* Prentice Hall, 1988.

[59] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes in C, Second Edition*, ch. 16, pp. 707–752. Cambridge University Press, 1992.

[60] D. E. Goldberg, *Genetic Algorithms in Search, Optimisation and Machine Learning.* Addison Wesley, 1989.

[61] A. H. Wright, *Foundations of Genetic Algorithms*, ch. Genetic Algorithms for Real Parameter Optimisation, pp. 205–218. Morgan Kaufmann, 1991.

[62] L. Booker, *Genetic Algorithms and Simulated Annealing*, ch. Improving search in genetic algorithms. Morgan Kaufmann, 1987.

[63] W. M. Spears and K. A. D. Jong, *Foundations of Genetic Algorithms*, ch. An Analysis of Multi-Point Crossover, pp. 301–315. Morgan Kaufmann, 1991.

[64] G. Syswerda, "Uniform crossover in genetic algorithms.," in *3rd International Conference on Genetic Algorithms (ICGA)*, pp. 2–9, 1989.

[65] R. A. Caruana, L. A. Eshelman, and J. D. Schaffer, "Representation and hidden bias ii: Eliminating defining length bias in genetic search via shuffle crossover," in *Eleventh International Joint Conference on Artificial Intelligence* (N. S. Sridharan, ed.), vol. 1, pp. 750–755, 1989.

[66] C. Z. Janikow and Z. Michalewicz, "An experimental comparison of binary and floating point representations in genetic algorithms," in *4th International Conference on Genetic Algorithms (ICGA)*, pp. 31–36, 1991.

[67] D. M. Tate and A. E. Smith, "Expected allele convergence and the role of mutation in genetic algorithms," in *5th International Conference on Genetic Algorthims (ICGA)*, pp. 31–37, 1993.

[68] T. C. Fogarty, "An incremental genetic algorithm for real-time learning," in *6th Int. Workshop on Machine Learning*, pp. 416–419, 1989.

[69] H. Mühlenbein, M. Schomisch, and J. Born, "The parallel genetic algorithm as a function optimiser," *Parallel Computing*, no. 17, pp. 619–632, 1991.

[70] T. Starkweather, D. Whitley, and K. Mathias, "Optimisation using distributed genetic algorithms," in *Parallel Problem Solving From Nature 1*, no. 496 in Lecture Notes in Computer Science, pp. 176–185, Springer-Verlag, 1990.

[71] H. Zhang and M. Ishikawa, "An extended hybrid genetic algorithm for exploring a large search space," in *2nd Int. Conf. Autonomous Robots and Agents*, (New Zealand), pp. 244–248, December 2004.

[72] A. J. Chipperfield, P. J. Fleming, and C. M. Fonseca, "Genetic algorithm tools for control systems engineering," in *Adaptive Computing in Engineering Design and Control*, (Plymouth Engineering Design Centre), pp. 128–133, 21-22 September 1994.

[73] R. G. Brown, "Engineering a beowulf-style compute cluster." Internet: `http://www.phy.duke.edu/~rgb/Beowulf/beowulf_book/beowulf_book/index.html`. Last Accessed: 19/04/06.

[74] J. Dongarra, "Faq on the linpack benchmark and the top 500 supercomputer sites." Internet: `http://www.netlib.org/utk/people/JackDongarra/faq-linpack.html`, 2004. Last Accessed: 19/04/06.

[75] J. Chaplin, "Cs2 wave simulator." Internet: `www.civil.soton.ac.uk/hydraulics/download/downloadtable.htm`. Last Accessed: 19/04/06.

[76] S. H. Salter, "Wave energy lecture notes." The University of Edinburgh.

[77] L. N. Patel, J. Hallam, and A. Murray, "Improved, simpler neural controllers for lamprey swimming," in *Proceedings of the International Conference of Artificial Neural Networks*, LNCS 3697, pp. 445–450, 2005.

Appendix A
System Simulator

Here we describe how we implement the numerical model for the mechanical and neural systems. The code has been written in C and this appendix describes how the simulator is constructed, highlighting the most significant sections and describes how this code was used to model the systems described in this thesis.

A.1 Organisation

The whole system is based around two independent sections, the mechanical WEC model and the model of the neural network. The separation of the two main sections was due to the fact that the separate systems were developed and verified independently; although the re-writing of these units to form a more homogeneous unit would have been preferable, the time required to do this was disproportionate to the improvement it may have offered.

The system can be divided into four distinct areas; Initialisation & program flow, the mechanical model, the neural model and communication.

Initialisation & Program Flow

Primarily this part of the program exists to co-ordinate the two numerical model routines and facilitate communication between them. As both of these models use different ODE solvers, the only way to allow data to be interchanged between them during the simulation is to divide up the overall simulation time into a number of 'chunks' then evaluate sequentially the mechanical and then the neural model over this small time chunk then exchange data between the two. This chunk-size is defined as command line option "-c [n]" and for a 60 second simulation a common value was 600. The arrangement of this control routine also allows for either the mechanical or neural models to be operated independently with the use of the "-o" or "-l" switch respectively

The Mechanical Model

Designed to be as expandable as possible the mechanical model evaluates the supplied equations of motion using a variable step Runge-Kutta ODE solver derived from one provided in numerical recipes [59]. This model evaluates the equations supplied in 'eqns.c' over a waveform that has been produced by one of the subroutines in excite.c. A variety of waveforms can be used, from swept waveforms to bichromatic and polychromatic waves to waves generated in a different program and imported, all of which can be specified on the command line. The initialisation and execution of the mechanical model is done in pelamis.c with the results appended to a global vector to be output to file later (output.txt).

The Neural Model

Developed with help from Leena Patel, these routines solve the equations that govern the neural output. These equations are solved using a simple fixed-step Runge-Kutta ODE solver developed from one presented in numerical recipes [59]. The potential of each neuron is calculated sequentially in a loop with the new value being based upon the sum of the potential of all the neurons[1] connected to this neuron, each multiplied by their synaptic weight. These synaptic weights are read in from an external file whose format is described below.

Communication

In order to pass information between the mechanical model and the neural model and vice-versa a small routine is called at the end of each major time-step (or 'chunk'). This routine take the current device motion and translates this into a sensory input value for the neuron model. It also take the current MN output from the neuron model and decides what damping value should be applied to the mechanical model.

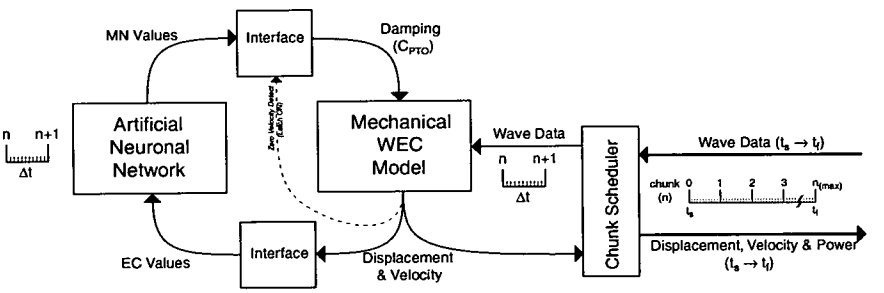


Figure A.1: *Block diagram of overall simulator operation.*

A.2 Operation

A.2.1 Command Line Options

The simulator can be invoked with the command '`simulate [options]`' these options are listed below. For example we can use the following command:

```
simulate -s 60 -c 600 -b -k -n[weights file] -d[damping file]
                            -f[wavefile]
```

[1] From the previous time-step

-s [duration] Simulation duration (seconds): Required.

-y [wavelength]

Used in conjunction with either regular or irregular wave generator, Specifies the wavelength of the generated wave: Must be used in conjunction with -p [period] and -h [height] and a waveform type specifier.

-f [filename]

Reads a file in from location specified in [filename] and uses this as the wave data applied to the model. The use of this switch requires that -r or -g are not specified. The format for this file is given in section A.2.2.

-r Random wave generator: Not fully implemented - see later releases.

-g

Regular wave generator: Generates a monochromatic wave with parameters specified by -y [wavelength] -p [period] and -h [height] for a period 10% longer than specified by -s [duration] and writes the output to `wavedata.txt`.

-p [period]

Used in conjunction with either regular or irregular wave generator, Specifies the period, in seconds, of the generated wave: Must be used in conjunction with -y [wavelength] and -h [height] and either -r or -g.

-h [height]

Used in conjunction with either regular or irregular wave generator, Specifies the amplitude of the generated wave in meters (peak to trough height will be twice this value): Must be used in conjunction with -y [wavelength] and -p [period] and must be specified with any waveform type generator.

-c [chunks]

This switch specifies the overall number of time-steps to split the simulation into. It does not affect the time-step size for the mechanical or neural model, however it does affect how often they can interchange information at this only happens at the end of each global time-step. A good starting value is 10× the duration(in seconds). the default value is 20.

-l

Simulate lamprey neural model only. With this option the mechanical model is ignored although the effect of the global time-step, '-c' does still affect the output.

-o

Simulate mechanical model only. With this option the neural model is ignored although the effect of the global time-step, '-c' does still affect the output.

-e [G_E]

Specifies the global excitation to be applied to the neural model. The default value is 0.3.

-x [E_E]

This specifies the amount of extra excitation applied to the neural model on top of the global excitation. Default is zero.

-w [start] [stop]

Generate a swept waveform with the starting and finishing periods specifies over the duration of the simulation. Note: there is a small regular section at the start and the finish of the generated waveform before the period sweep takes place. This can also be combined with -a to sweep the wavelength as well as the period.

-a [start] [stop]

Normally used in conjunction with -w but possible to use independently, this option produces a waveform whose period sweeps from the specified start to the finish over the duration of the simulation.

-m [p1] [p2] [p3]

Very basic polychromatic waveform generator that produces a waveform comprising a spectrum of the three periods specified in p1, p2 and p3. Assumes the phase of these periods is all zero. Must be specified with -h for wave-height.

-v

Specifies verbose output. Without this switch the programs' only output is the total power developed written to the command line. Specify this option in order to get any detailed data from the program. Very useful option for batch processing as the program can generate quite large files.

-d [filename]

Loads the damping values for the mechanical simulation in from *filename*. The format of this file is specified below in section A.2.2.

-n [filename]

Loads the neural weights in from an external file whose location is specified by *filename*. The format of this file is specified below in section A.2.2.

-b

Evaluate using a selection of four pre-selected monochromatic waveforms. These are specified as wavedata1, wavedata2, wavedata3.txt and wavedata4.txt Which can comprise any waveform in the correct format (see below). The system under test is evaluated in each of these waveforms sequentially, 1 through to 4, and the average power across all returned.

-i

Normalise the output power with -b, not implemented fully. Works only for one specific case - do not use.

-k

Implements practical latching. Without this switch full latching is implemented whereby latch-on **and** latch-off points are determined by the MN output. With this option the output is automatically latched when velocity equals zero and the MN output determines **only** the latch off point. Can be considered a REQUIRED option unless very specific testing is required.

A.2.2 File Formats

A.2.2.1 wave data file

This file specifies the water height against time and is tab separated.

Any lines preceded by a hash (#) are considered comments.

Each file should have one line preceded by an asterix (*) that defines the waveheight. It should be in the format:

[time(s)] [water level] [velocity ($= \frac{\lambda}{period}$)]

below is a section of file generated by the program:

```
#Pelamis - Lamprey simulation WAVE DATA FILE
#All program code copyright Dr Leena Patel & Tim Mundon 2003
1.000000
0.000000    0.000000    0.645161
0.010000    0.020267    0.645161
0.020000    0.040526    0.645161
0.030000    0.060768    0.645161
0.040000    0.080985    0.645161
```

```
0.050000      0.101168      0.645161
0.060000      0.121311      0.645161
0.070000      0.141403      0.645161
.....
.....
EOF
```

A.2.2.2 damping file

This file specifies the damping to be applied to the mechanical system. It is a very simple file and consists of only three values on a single line that must be preceded by a space. Comments are allowed if preceded by a hash (#). If a single buoy system is simulated then just the first value needs to be specified.

format:
`<spc>[value 1]<spc>[value 2]<spc>[value 3]`

A.2.2.3 neural weights file

This file specifies the synaptic weights to be applied to the neural model. This file is organised into distinctive sections that represent the different types of connections. The format shown here illustrates the how the file relates to the particular synapses. The following representation is used:

MN, LIN, EIN, CCIN:	Neuron type.
BI:	Braininput.
EC:	Edge cell.
(Neuron type)L:	Left side neuron *e.g. LINL=left side LIN neuron*
(Neuron type)R:	Right side neuron *e.g. EINR=right side EIN neuron*
(Neuron type)L/Ra:	Neuron in adjacent segment *e.g. EINRa=right side EIN neuron in adjacent segment*
braininput:	Identifies the connections that extend from the brain into the neurons.
ec-weights:	Identifies the sensory input.
pconnxns:	Excitatory connections between neurons.
nconnxns:	Inhibitory connections between neurons.
prostral:	Excitatory forward connections between segments.
pcaudal:	Excitatory rearward connections between segments.
nrostral:	Inhibitory forward connections between segments.
ncaudal:	Inhibitory rearward connections between segments.
(neuron)-(neuron):	Denotes direction of connections [from]-[to].

Note: The network is assumed to be symmetrical so these values will be mirrored to produce the other half of the network.
Note: To remove all interconnections between segments the values of n/pcaudal and n/prostral must still be included and be set to zero.

```
braininput
BI-MN,BI-LIN,BI-EIN,BI-CCIN
#
ec_weights
EC-MN,EC-LIN,EC-EIN,EC-CCIN
#
pconnxns
MNL-MNL,LINL-MNL,EINL-MNL,CCINL-MNL
MNL-LINL,LINL-LINL,EINL-LINL,CCINL-LINL
MNL-EINL,LINL-EINL,EINL-EINL,CCINL-EINL
MNL-CCINL,LINL-CCINL,EINL-CCINL,CCINL-CCINL
MNL-MNR,LINL-MNR,EINL-MNR,CCINL-MNR
MNL-LINR,LINL-LINR,EINL-LINR,CCINL-LINR
MNL-EINR,LINL-EINR,EINL-EINR,CCINL-EINR
MNL-CCINR,LINL-CCINR,EINL-CCINR,CCINL-CCINR
#
nconnxns
..
same format as pconnxns
..
#
prostral
MNL-MNLa,LINL-MNLa,EINL-MNLa,CCINL-MNLa
MNL-LINLa,LINL-LINLa,EINL-LINLa,CCINL-LINLa
MNL-EINLa,LINL-EINLa,EINL-EINLa,CCINL-EINLa
MNL-CCINLa,LINL-CCINLa,EINL-CCINLa,CCINL-CCINLa
MNL-MNRa,LINL-MNRa,EINL-MNRa,CCINL-MNRa
MNL-LINRa,LINL-LINRa,EINL-LINRa,CCINL-LINRa
MNL-EINRa,LINL-EINRa,EINL-EINRa,CCINL-EINRa
MNL-CCINRa,LINL-CCINRa,EINL-CCINRa,CCINL-CCINRa
#
pcaudal
..
same format as prostral
..
#
nrostral
..
same format as prostral
..
#
ncaudal
..
same format as prostral
..
EOF
```

A.2.2.4 `output.txt`

This file is the primary output from the program. It consists of 23 tab separated columns of data, these are as follows:

- *Column 1*: Time (seconds).
- *Column 2*: Displacement of buoy 1.
- *Column 3*: Velocity of buoy 1.
- *Column 4*: Displacement of buoy 2 *(if used).*
- *Column 5*: Velocity of buoy 2 *(if used).*
- *Column 6*: Displacement of buoy 3 *(if used).*
- *Column 7*: Velocity of buoy 3 *(if used).*
- *Column 8*: Displacement of buoy 4 *(if used).*
- *Column 9*: Velocity of buoy 4 *(if used).*
- *Column 10*: Relative velocity of buoy 1 & 2.
- *Column 11*: Relative velocity of buoy 2 & 3.
- *Column 12*: Relative velocity of buoy 3 & 4.
- *Column 13*: Relative Displacement of buoy 1 & 2.
- *Column 14*: Relative Displacement of buoy 2 & 3.
- *Column 15*: Relative Displacement of buoy 3 & 4.
- *Column 16*: Instantaneous Power from buoy 1 & 2 (or just from buoy 1 if a single buoy system).
- *Column 17*: Instantaneous Power from buoy 2 & 3.
- *Column 18*: Instantaneous Power from buoy 3 & 4.
- *Column 19*: Damping for buoy 1 & 2 (or just for buoy 1 if a single buoy system).
- *Column 20*: Damping for buoy 2 & 3.
- *Column 21*: Damping for buoy 3 & 4.
- *Column 22*: Total instantaneous power.
- *Column 23*: Input wave.
- *Column 24*: Acceleration of buoy 1. *not implemented fully*

A.3 Code

The source code for this simulator is included in the attached CD-ROM. For simplicity and repeatability the code is divided into three seperate units; the single buoy, two buoys, and three buoys/two neural segments. This can be found in the following locations:

cdroot/single_buoy/
cdroot/two_buoys/
cdroot/three_buoys/

These locations also include examples of all the apropriate data files.
NOTE: All code on the atached disk is for review purposes only. If any sections of this code are used in any other application then written permisson of the author & Edinburgh University MUST be saught beforehand.

Appendix B
Genetic Optimisation and Multiprocessor Use

The evolutionary computation within this thesis is implemented in MATLAB using the GA Toolbox [72]. It was found that this was a particuarly effective and efficient method of applying a genetic algorithm to the optimisation problem presented here and i would heartily recomend this as an excellent development tool. The GA Toolbox is a 3rd party add-on toolbox for MATLAB and its routines encompass all common genetic methods, as such for further information on any of the high-level GA commands presented here the reader is directed at chapter 4 and at the appropriate references.

This appendix includes the GA routines used at various stages in this thesis, the code that employs the multi-processing and the instructions so that the reader can run any of the examples presented.

B.1 Multi-Processing

All the GA routines presented here in section B.2 implement a multi-processing method to speed up the execution. They use the taskfarmer piece of code described in section B.3 to pass jobs out to a selection of machines for processing. In order to understand how the genetic algorithms presented here work it is prudent to first explain how the taskfarmer program operates.

Written in Perl, Taskfarmer contains two elements, a server and a client. The server program maintains a list of remote hosts and once it recieves an instruction to evaluate an individual it opens an SSH connection to the remote machine, this starts the remote client program which executes the requested script. This program is designed to work in a common filesystem enviroment such as NFS where all machines have access to the same user filespace.

Taskfarmer simply initiates a command (to run a script) on a remote computer, the result of this script is then written as a file to a common area. In the context of this application, in order to evaluate a sinlge generation of individuals the GA turns each individual into two seperate numbered files (`weights.n.txt` & `damping.n.txt` where n is an integer) in the common file area. The GA then writes a 'job file', where each line of this file details the command that is to be run on the remote machine. This file is then passed to the server routine, `tfSSHserver.pl` which must be running on the current machine, using the `tfAdd.pl` script. The server then opens an SSH tunnel to as many machines as are entries in the job file upto the maximum number of machines avaliable (these are listed in `hosts.txt`). This

SSH tunnel triggers a client program to execute the required command on the host, in this case this script is `startsimulator` which executes the simulator (located in the common area) using an appropriate weights & damping file (written to the common area earlier) and writes the result to another file in the common area called `resultn` where n is an integer and relates to the number of the weight & damping file. The GA waits for the remote clients to finish by watching for the creation of the result files and then reads in the data they contain.

In practice NFS can be quite slow if a number of machines are trying to access the same space conncurrently so the first call to taskfarmer uses a slightly different script, `startsimulator.1st`, that copies the simulator code to the remote machines' local hard disk. Each following call to taskfarmer uses `startsimulator` which executes the simualator on the machines local disk but writes to and reads from the common file ares, ensuring that the GA can access the results. This adjustment increased speed over 100% compared to running the simulator code from the shared file area.

B.2 GA routines

As an example, the MATLAB code for the GA is shown below. This was used with the single buoy experiments to evolve solutions. This and the other GA routines used are included in the attached CD-ROM.

```
function output=evolve(Ttm)

%assmume taskfarmer server is running
%This routine MUST be run on pclxa03 or another vlx/tlc workstation

%define data size
NIND = 40;                    % Number of individuals
MAXGEN = 1000;                 % initial Maximum no. of generations
%start stopwatch timer
tstart = clock;
pause on

fid = fopen('~/PhD_Work/Processing/output/ga_results.txt','w');
if fid >0
  fprintf(fid,'\nsimulation started on:');
  fprintf(fid,' %g %g/%g %g:%g %gs\n',fix(clock));
  fclose(fid);
end

NVAR = 73;                      % No. of variables (including damping)
PRECI = 7;                   % Precision of variables
GGAP = 0.9;                     % Generation gap
count = 1;                    % Initial generation monitor
cint = 1;                     % Initial generation monitor
mutprob = 0.5;                % Extra mutation rate  increased from 0.4 -> 0.5
mutcount = 10;

%define evaluation vector
ObjV=zeros(NIND,1);

% Build field descriptor
```

```
FieldDR = rep([0;15],[1 NVAR]);
%FieldD = [rep([PRECI],[1,NVAR]);...
%rep([0;15],[1,NVAR]); rep([1;0;1;1],[1,NVAR])];

% Initialise population
Chrom = crtrp(NIND, FieldDR);
%Chrom = crtbp(NIND, NVAR*PRECI);
gen = 1; % Counter

% Evaluate initial population
%INDS = bs2rv(Chrom,FieldD);
INDS = Chrom;

%evaluate initial population
%farm out jobs
for i=1:NIND
    %take rows of input data
    dest_data = INDS(i,:);
    %clear up any resident results
    unix(['rm ~/PhD_Work/Processing/result' num2str(i)]);

    % compile commands & filenames
    filename=strcat('weights.',num2str(i),'.txt');
    dampname=strcat('damping.',num2str(i),'.txt');

    % write scriptfile
    if i==1
       unix(['echo ./startsimulator.1st -a' filename ' -b' dampname ' -cresult' num2str(i)
    else
       unix(['echo ./startsimulator.1st -a' filename ' -b' dampname ' -cresult' num2str(i)
    end

    %Prune weights such that <1 = 0
    %for b=2:73
    %   if dest_data(b) < 1
    %       dest_data=0;
    %   end
    %end
    % write chrom to file:
    feval(@writefile,dampname,filename,dest_data);

end

%Do computation on pclxa03.see.ed.ac.uk
%Process GA_script and simulate individuals
unix('tfAdd.pl pclxa03.see.ed.ac.uk 3335 h ~/PhD_Work/Processing/GA_script');

%collect results
for i=1:NIND
    %detect result
    i
    eval_t='';
    fid=fopen(['~/PhD_Work/Processing/result' num2str(i)],'r');
    while strcmp(eval_t,'')
      if fid<0
         fid=fopen(['~/PhD_Work/Processing/result' num2str(i)],'r');
         pause(0.5);
      else
```

```
        eval_t = fscanf(fid, '%c');
      end
    end

    fclose(fid);
    [m,n]=size(eval_t);
    if m<1;
       eval_t='0';
    end
    eval_v = str2num(eval_t);
    %clear up
    unix(['rm ~/PhD_Work/Processing/result' num2str(i)]);

    ObjV(i)= eval_v * -1;

    % monitor
    fid = fopen('~/PhD_Work/Processing/output/ga_results.txt','a');
    fprintf(fid,'#');
    fclose(fid);

end

% Generational loop
while gen < MAXGEN,

        %mark generation in results file
        fid = fopen('~/PhD_Work/Processing/output/ga_results.txt','a');
        if fid >0
            fprintf(fid,'\n\nGeneration %g',[gen]);
            fclose(fid);
        end

        %read (update) maximum generations
fid = fopen('~/PhD_Work/Processing/maxgen','r');
        if fid >0
            MAXGEN = fscanf(fid,'%g');
            fclose(fid);
        end

        % Assign fitness values to entire population
        FitnV = ranking(ObjV);

        % Result monitor%
        if isequal(gen,count)
            count = count + cint;
            intav = mean(ObjV);
            intObjV = ObjV * -1;
            intFitnV = ranking(intObjV);
            [r,index] = sort(intFitnV);

            %write best individual to file
            fid = fopen('~/PhD_Work/Processing/output/ga_results.txt','a');
            if fid >0
                fprintf(fid,'\n#1 individual fitness %g\n', ObjV(index(1)));
                fprintf(fid,'\noutput file no: %g.\n',index(1));
                fclose(fid);
                cpcmd=['cp ~/PhD_Work/Processing/weights.' num2str(index(1)) '.txt ~/PhD_
unix(cpcmd);
```

```
            end
        end

        % Select individuals for breeding
        SelCh = select('sus', Chrom, FitnV, GGAP);

        % Recombine individuals (crossover)
        SelCh = recombin('recint',SelCh,0.7); %intermediate recomb. (real value)
        %SelCh = recombin('xovsp',SelCh,0.7); %single-point crossover (xovmp=multipoint)

        % Apply mutation
        if isequal(gen,mutcount)
            mutcount=mutcount+10;
            SelCh = mutbga(SelCh,FieldDR,mutprob);
            %SelCh = mut(SelCh,mutprob);
        end
SelCh = mutbga(SelCh,FieldDR);
        %SelCh = mut(SelCh);

        % Calculate size of population
        [nind,length]=size(SelCh);
        ObjVSel=zeros(nind,1);

        %Re-insert offspring into population (modification)
        %Chrom = reins(Chrom,SelCh,1,1,ObjV);

        %convert new offspring
        %INDS = bs2rv(SelCh,FieldD);
        INDS = SelCh;
        %INDS = Chrom;

        % Evaluate Population:
%farm out jobs
for i=1:nind
    %take rows of input data
            dest_data = INDS(i,:);

            % compile commands & filenames
            filename=strcat('weights.',num2str(i),'.txt');
            dampname=strcat('damping.',num2str(i),'.txt');

            % write scriptfile
    if i==1
                unix(['echo ./startsimulator -a' filename ' -b' dampname ' -cresult'
 num2str(i) ' > ~/PhD_Work/Processing/GA_script']);
            else
        unix(['echo ./startsimulator -a' filename ' -b' dampname ' -cresult'
 num2str(i) ' >> ~/PhD_Work/Processing/GA_script']);
            end

    %Prune weights such that if<1 = 0
    %for b=2:73
    %   if dest_data(b) < 1
    %       dest_data=0;
            %   end
    %end

    %write chrom to file
```

```
            feval(@writefile,dampname,filename,dest_data);
        end

        %Do computation on server running on pclxa03.see.ed.ac.uk
        %Process GA_script and simulate individuals
        unix('tfAdd.pl pclxa03 3335 h ~/PhD_Work/Processing/GA_script');

        %collect results
        for i=1:nind
            %detect result
            eval_t='';
            fid=fopen(['~/PhD_Work/Processing/result' num2str(i)],'r');
            while strcmp(eval_t,'')
               if fid<0
                   fid=fopen(['~/PhD_Work/Processing/result' num2str(i)],'r');
                   pause(0.5);
               else
                   eval_t = fscanf(fid, '%c');
               end
            end

            fclose(fid);
            [m,n]=size(eval_t);
            if m<1;
                eval_t='0';
            end
            eval_v = str2num(eval_t);

            %clear up
            unix(['rm ~/PhD_Work/Processing/result' num2str(i)]);

            %ObjV(i)= eval_v * -1; %whole population is evaluated (rather than just
the new individuals)
            ObjVSel(i)= eval_v * -1;

            % monitor
            fid = fopen('~/PhD_Work/Processing/output/ga_results.txt','a');
            fprintf(fid,'#');
            fclose(fid);
        end

        % Reinsert offspring into population
        [Chrom ObjV]=reins(Chrom,SelCh,1,1,ObjV,ObjVSel);

        % Increment counter
        gen = gen+1;

end

%calculate length of simulation
elapsed = etime(clock, tstart);

% Select Best individuals

ObjV = ObjV * -1;
FitnV = ranking(ObjV);
[r,index] = sort(FitnV);
%Phen = bs2rv(Chrom, FieldD);
```

```
Phen = Chrom;
average = mean(ObjV);

fprintf('\nElapsed time for simulation: %gs.',elapsed);
fprintf('\noutput file no: %g.',index(1));

output.rank=1;
output.values=Phen(index(1),:);
output.result=ObjV(index(1));
output.mean=average;

%copy best individual
cpcmd=['cp ~/PhD_Work/Processing/weights.' num2str(index(1)) '.txt ~/PhD_Work/Proces
sing/output/best_individual_weights.' num2str(gen) '.txt'];
unix(cpcmd);
unix('rm maxgen');

%output stats to file
fid = fopen('~/PhD_Work/Processing/output/ga_results.txt','a');
if fid >0
    fprintf(fid,'\nSimulation finished. Elapsed Time = %gs', elapsed);
    fprintf(fid,'\n#1 individual fitness value: %g',ObjV(index(1)));
    fprintf(fid,'\npopulation average: %g',average);
    fprintf(fid,'\noutput file no: %g.',index(1));
    fclose(fid);
end
return


function output = writefile(dampname, filename, i_input)
%function that will take a 201 variable string and write it to a textfile
%for use when evolving the weights & damping
    %damping file
    fid = fopen(dampname,'w');
    val=i_input(1)*1000; %scale damping value to be between 0 and 15,000
    fprintf(fid,' %6.0f 10000000 10000000\n', val);
    fclose(fid);
    %weights file
    fid = fopen(filename,'w');

    fprintf(fid,'braininput\n%g,%g,%g,%g\n#\n', i_input(2:5));
    fprintf(fid,'ec_weights\n%g,%g,%g,%g\n#\n', i_input(6:9));
    fprintf(fid,'pconnxns\n');
    for n=0:7
        fprintf(fid,'%g,%g,%g,%g\n', i_input((n*4)+10:(n*4)+13));
    end
    fprintf(fid,'#\nnconnxns\n');
    for n=0:7
        fprintf(fid,'%g,%g,%g,%g\n', i_input((n*4)+42:(n*4)+45));
    end
    fprintf(fid,'#\nprostral\n');
    for n=0:7
        fprintf(fid,'0,0,0,0\n');
        %fprintf(fid,'%g,%g,%g,%g\n',i_input((n*4)+74:(n*4)+77));
    end
    fprintf(fid,'#\npcaudal\n');
    for n=0:7
        fprintf(fid,'0,0,0,0\n');
```

```
        %fprintf(fid,'%g,%g,%g,%g\n', i_input((n*4)+106:(n*4)+109));
    end
        fprintf(fid,'#\nnrostral\n');
    for n=0:7
        fprintf(fid,'0,0,0,0\n');
        %fprintf(fid,'%g,%g,%g,%g\n', i_input((n*4)+138:(n*4)+141));
    end
    fprintf(fid,'#\nncaudal\n');
    for n=0:7
        fprintf(fid,'0,0,0,0\n');
        %fprintf(fid,'%g,%g,%g,%g\n', i_input((n*4)+170:(n*4)+173));
    end
        fprintf(fid,'EOF');
    fclose(fid);
return ;
```

B.2.1 Startsimulator Routine

Note: This routine assumes that the simulator is present on the local machine. This is achieved by running `startsimulator.1st` routine beforehand, this is identical to the routine below except that it copies the simulator to the local machine before executing it.

```
#!/bin/bash
#this file must be located in ~/
#executable must be located in /home/trm/PhD_Work/Processing/lamp-pel/
#neural weights files must be located in ~/PhD_Work/Processing/
#output will be located in directory ~/PhD_Work/Processing/

#Change to 'simulate' directory
cd /tmp/lamp-pel_tmp/;

#Update executable
cp ~/PhD_Work/Processing/lamp-pel/simulate /tmp/lamp-pel_tmp/

#Setup getopts to process command line arguments
set -- `getopt "a:b:c:" "$@"`

#Process arguments and execute 'simulate'
while [ ! -z "$1" ]
  do
  case "$1" in
      -a) ./simulate -s 60 -c 600 -k -n ~/PhD_Work/Processing/$2 -d ~/PhD_Work/P
rocessing/$4 -f wavedata2.txt > ~/PhD_Work/Processing/$6;;
      *) break;;
  esac

#display the executed command line
  echo "./simulate -s 100 -c 1000 -k -b -n ~/PhD_Work/Processing/$2 -d ~/PhD_Wor
k/Processing/$4 > ~/PhD_Work/Processing/$6;"

  shift
done
$
```

B.3 Taskfarmer Routines

The following sections show the perl scripts used to pass the GA to multiple processors.

B.3.1 Server Routine

```
#!/usr/bin/perl

use IO::Socket;

sub addJobs {
  my ($Script) = @_;

  @NewJobs = split /\n/,`cat $Script`;

  print "scripts =";
  print "@NewJobs\n";

  splice (@JobList, $#JobList+1, 0, @NewJobs);

  $numJobs += $#NewJobs + 1;

}

sub addJobsHead {
  my ($Script) = @_;

  @NewJobs = split /\n/,`cat $Script`;

  $numJobs += $#NewJobs + 1;

  splice (@NewJobs, $#NewJobs+1, 0, @JobList);

  @JobList = @NewJobs;
}

sub startClients{
  my (@Clients) = @_;
  foreach $node (@Clients){

    #print "ssh -x -o BatchMode=yes -n $node ~/bin/tfClient.pl $HOST $PORT &\n";
    system("ssh -x -o BatchMode=yes -n $node ~/bin/tfClient.pl $HOST $PORT &");

    print "Starting client on $node.\n\n";

  }
}

#Check for arguments
if($#ARGV+1 == 1){
    $hangup = 0;
    print "No Hang Up!\n";
} else {
    $hangup = 1;
}
```

```
#Get the port number from the command line.
$PORT = @ARGV[0];

#Set the host to be the computer the script is running on.
$HOST = $ENV{'HOSTNAME'};

#Set HOSTFILES to the directory containing the 'clusterList' file.
$HOSTFILES = "/home/trm/taskfarmer/";

#The path to the log file.
$LOG = "./tfLog.txt";

#Clear log file.
open(LOGHANDLE, "> $LOG");
close(LOGHANDLE);

#Get the node list
#The same node may appear in the list more than once, with the number of
#appearances equal to the number of processes that may run on that node.

@NODELIST = split /\n/,`cat $HOSTFILES/clusterList`;

#Get the job list
for($i=1; $i<$#ARGV+1; ++$i){

    print"@ARGV[$i]\n";
    addJobs(@ARGV[$i]);
}
$numJobs = $#JobList + 1;

#Create a socket
#The que can hold a single connection from each node.
my $sock = new IO::Socket::INET (
                                    LocalHost => $HOST,
                                    LocalPort => $PORT,
                                    Proto => 'tcp',
                                    Listen => $#NODELIST+5,
                                    Reuse => 1,
                                   );
print "host sock = $sock\n\n";
die "Server - Could not create socket: $!\n" unless $sock;

# set buffering off
$sock->autoflush(1);

#Start a tf client running on each node
startClients(@NODELIST);
print "here\n\n";
#Now process connections from client nodes.
my ($new_client, $buf);

$started = 0;
$returned = 0;

$failed = 0;
  print "Here1\n";
while(($started == 0 || $started != $returned) || !$hangup) { # wait for and accept a conn
  print "Here\n";
```

```
  $new_client = $sock->accept();
  print "Server - connection received.\n";

  #get ip of remote machine and translate to a name
  $new_client_name = gethostbyaddr($new_client->peeraddr, AF_INET);

  #clientName is assigned name but if that fails use ip address instead
  $clientName = $new_client_name || $new_client->peerhost();

  $buf = <$new_client>; # read from the socket
  chomp $buf;

  print "buf = $buf\n";

  $RealRequest = 0;

  if($buf eq "FIRST"){
    #Just issue a new job
      print "Client request 'first job' of /$numJobs: $clientName\n";
       $RealRequest = 1;
  } else {
    if($buf =~ s/^SUCCESS//){
      ++$returned;
      open(LOGHANDLE, ">> $LOG");
      print LOGHANDLE "$returned - Success: $clientName $buf\n\n";
      close(LOGHANDLE);
      print "Client returning 'successful job' $returned/$numJobs : $clientName\n";
      $RealRequest = 1;
    } else {
      if($buf =~ s/^FAILED//){
++$returned;
open(LOGHANDLE, ">> $LOG");
print LOGHANDLE "$returned - Failure: $clientName $buf\n\n";
close(LOGHANDLE);
++$failed;
print "Client returning 'failed job' $returned/$numJobs : $clientName\n";
$RealRequest = 1;
      } else {
if($buf =~ s/^NEWSCRIPTS//){

  if($buf =~ s/^HEAD//){
    $head = 1;
  } else {
    if($buf =~ s/^TAIL//){
      $head = 0;
    } else {
      $head = -1;
    }
  }

  if($head != -1){

    @NewScripts = split ":", $buf;

    foreach $NewScript (@NewScripts){

      if(-e $NewScript){
```

```
#Add new jobs
if($head == 1){
  addJobsHead($NewScript);
} else {
  addJobs($NewScript);
}

open(LOGHANDLE, ">> $LOG");
print LOGHANDLE "New jobs added from script \"$NewScript\".\n\n";
close(LOGHANDLE);

print "New jobs added from script \"$NewScript\".\n\n";

#Restart clients that have closed
startClients(@RestartList);

#Empty the restart list;
@RestartList = ();

      }
    }
  } else {
    open(LOGHANDLE, ">> $LOG");
    print LOGHANDLE "New jobs not added. Poorly formed queue position.\n\n";
    close(LOGHANDLE);
  }
}
      }
    }
  }

  if( $RealRequest == 1){
    #Assign new job
    if($#JobList+1 > 0){
      $job = shift(@JobList);
      print $new_client "$job\n";
      ++$started;
      print "\tNew job issued. $started\n\n";
    } else {
      #No more jobs. Close the client.
      print $new_client "DONE\n";
      print "\tNo jobs left. Client closed.\n\n";

      #Push client onto a list for possible restart if more jobs arrive.
      push(@RestartList, $clientName);

    }
  } else {
    print "No request made. Socket closed\n";
  }
}
close($sock);

open(LOGHANDLE, ">> $LOG");
print LOGHANDLE "\tTotal Failures = $failed.\n";
close(LOGHANDLE);
$
```

B.3.2 Client Routine

```
#!/usr/bin/perl

use IO::Socket;

$MaxFails = 3;

sub get_socket {
  #Report to the server for a job
  my $sock = new IO::Socket::INET (
   PeerAddr => $Server,
   PeerPort => $Port,
   Proto => 'tcp',
  );
  print "client socket = $sock\n";
  die "Client - Could not create socket: $!\n" unless $sock;

  # set buffering off
  $sock->autoflush(1);
  return ($sock);
}

#*************************************************************
#Main Prog.
#*************************************************************

#Check for arguments
if($#ARGV+1 != 2){
  die "You must give the server address and port number.\n tfClient serverAddr portNumber.
}

$Server = $ARGV[0];
$Port = $ARGV[1];

$sock = get_socket;

#Request first job.
print $sock "FIRST\n";

$job = <$sock>;
chomp $job;
close($sock);

$Failed = 0; #Count number of failed attempts at running a job

while($job ne "DONE"){

  #2>&1 redirect stderr to stdout
  $timingInfo = `bash -c '/usr/bin/time 2>&1 -f "%Uuser %Ssystem %Eelapsed %PCPU" $job >>

  $exitStatus = $?;

  if($exitStatus != 0){
    if($Failed < $MaxFails){
      #Job Failed so try again.
      ++$Failed;
      next;
```

```
    } else {
      #Job has completely failed. Tell the server.
      $sock = get_socket;
      print $sock "FAILED $job - $exitStatus\n";

    }
  } else {
    #Job has succeeded. Tell the server
    $Failed = 0;#Reset the failure counter.

    $sock = get_socket;

    print $sock "SUCCESS $job - $timingInfo\n";
  }
  #Get next job
  $job= <$sock>;
  close($sock);

  chomp $job;
}$
```

Appendix C
Causal Neural Control of a Latching Ocean Wave Point Absorber

Causal Neural Control of a Latching Ocean Wave Point Absorber

T.R. Mundon[1], A.F. Murray[1], J. Hallam[2], and L.N. Patel[1]

[1] Edinburgh University, The Kings Buildings, Edinburgh, UK
tim.mundon@ed.ac.uk
[2] University of Southern Denmark, Campusvej 55,
DK-5230 Odense M, Denmark

Abstract. A causal neural control strategy is described for a simple "heaving" wave energy converter. It is shown that effective control can be produced over a range of off-resonant frequencies. A latching strategy is investigated, utilising a biologically inspired neural oscillator as the basis for the control.

1 Introduction

It is vital that the energy retrieved from the ocean by a *Wave Energy Converter (WEC)* be maximised and much work has been done in this area [4]. However, due to the inherent unpredictability of the future wave, many advanced techniques that can implement near-optimum transfer require knowledge of the sea state immediately prior to reaching the WEC device. Our system uses a phase locked neural oscillator that tracks and optimises the motions of a simple point absorber WEC. A system with only one degree of freedom, with motion constrained to the vertical direction, is studied as a simplified exemplar of a wider class of WEC's. A biological system, the lamprey [2], provided the inspiration for the artificially evolved neural controller presented here. This system is implemented using a neural network in order to optimise the power generated over a range of frequencies and for a variety of input waveforms. A time domain based system was developed that does not require explicit prior knowledge of the input sea state. This approach effectively solves the equations of motion and the neural equations in parallel as separate, mutually dependent systems.

One of the fundamental requirements for efficiency in WEC's is that the correct phase must be maintained at frequencies away from resonance [1]. The primary method of phase control studied here is "latching" control [5,6,8], which has been shown to yield significant increases in power. Latching control provides a pseudo-resonant system by clamping the device rigid at the extremities of its motion until the wave force has increased to an optimum. The device is then released and thus generates power until reaching the next extremity of excursion. Although latching has so far been difficult to implement in irregular waves we outline a system that points towards a practical latching strategy for irregular waves that does not require explicit knowledge of the future state of the ocean.

W. Duch et al. (Eds.): ICANN 2005, LNCS 3697, pp. 423–429, 2005.

2 The Wave Energy Converter (WEC) Model

Advanced models of ocean wave energy converters can take account of complex wave-body interactions, including non-linear effects in order to model exactly how to absorb maximum energy from their environment. Whilst these models are essential to develop increased overall power capture, their inherent complexity makes them unsuitable for concept-proving experiments such as this. In order to better illustrate the method of control proposed, a point absorber, restricted to a single degree of movement in the vertical plane is considered. Experience with full WEC models suggests that this simplification captures most of the important characteristics of a more complex WEC model.

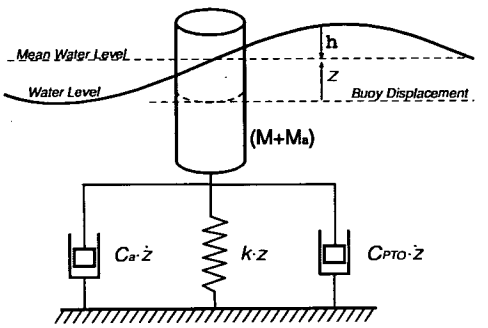


Fig. 1. Mechanical model of heaving buoy

Consider the simple mechanical system described in fig. 1, consisting of a single buoy constrained to oscillate in heave mode (i.e. vertically) only. The float is a cylinder of radius 1.65m and length 5m that is 50% submerged at equilibrium. The power take-off is represented as a damper, C_{PTO}. This simple harmonic oscillator system is described by equation 1.

$$F_{e(t)} = (M + M_a)\ddot{z} + C_a\dot{z} + C_{PTO}\dot{z} + kz \tag{1}$$

$$displacement = \eta - z \tag{2}$$

$$F_{e(t)} = \eta(\pi r^2 \rho_w g) \quad \& \quad kz = z(\pi r^2 \rho_w g) \tag{3}$$

The hydrodynamic co-efficients of added mass (M_a) and added damping (C_a) are defined by the buoy's geometry and are frequency dependent. As we aim to construct a control strategy requiring no foreknowledge of future waves, these have simply been approximated to static values. Although we appreciate that this is not strictly accurate it was found that such an approximation was adequate for this model.

In order to maintain a linear approximation, the force on the buoy is decomposed into the wave force F_e and the spring force kz which are proportional to water level(η) and buoy displacement(z) respectively. This is shown in eqn. 3, where $(\pi r^2 \rho_w g)$ is the restoring force on the cylinder.[3]

As power is the rate of energy absorbed in damper C_{PTO}, and given that from eqn. 1, $F_{(t)} = C_{PTO} \cdot \dot{z}$, it follows that: $P_{(t)} = F_{(t)} \cdot \dot{z} = C_{PTO} \cdot \dot{z}^2$

2.1 Existing Control Methods

Maximum power transfer between the wave and the device will occur when the natural period of these coincides, i.e resonance. Budal [1] found that it is the phase difference of $\frac{\pi}{2}$ between the forcing component and the device displacement that characterises this point of maximum power transfer. Away from resonance this phase difference reduces or increases. However, by varying the value of C_{PTO} in eqn. 1, it is possible to adjust the WEC's response at off-resonant periods, increasing the phase difference so as to optimise the power captured. However this is still an uncontrolled system as C_{PTO} remains constant over many periods, hence is is classed as "optimal real" damping in this paper (see fig. 3). Due to the unpredictable nature of sea waves, increases in off-resonance performance are desirable so that the effective bandwidth can be as wide as possible.

Many approaches to optimising the power developed with WEC's have been proposed and are reviewed in [4]. Latching, the simplest and hence perhaps the most significant of these methods, was initially proposed independently in [6] and [5] and then later in [8]. Latching enforces the correct phase shift between the water level and device displacement by locking the device at the extremities of its oscillatory cycle. This is implemented by locking the device displacement when velocity, $\dot{z} = 0$ and releasing it a certain time T_L later. The optimum power achievable from this method is intrinsically non-continuous, although it can be calculated iteratively [8]. Alternatively, reactive control [4] involves application of forces to the device that are in phase with both displacement and acceleration. Both latching and reactive control can develop significantly improved off-resonance power by using knowledge of the incoming wave.

3 A Neurally-Inspired Solution

In order to appreciate the inspiration for the control method proposed here, we first look at the articulated (snake-like) configuration of some real WEC devices. Generally consisting of two or more floats, it can be seen that these devices generate power through the relative motions of their individual sections.[10]

It can be further seen that this motion is similar to the propulsive anguiliform movement shown in many aquatic organisms. In general, it can be seen that this motion is controlled through a local neural system known as a "Central Pattern Generator" *(CPG)*. One particular vertebrate to demonstrate this locomotion, the lamprey, has a very well documented CPG structure and has been shown [7] to have a surprisingly simple and elegant neural architecture.

The structure of the lamprey CPG consists of a series of neural oscillators *(segments)*, each responsible for controlling a single section of the body. Interconnections between adjacent segments produce a small phase difference such that a travelling wave propagates along the length of the body. This overall motion is subject to modulation via sensory interaction [9].

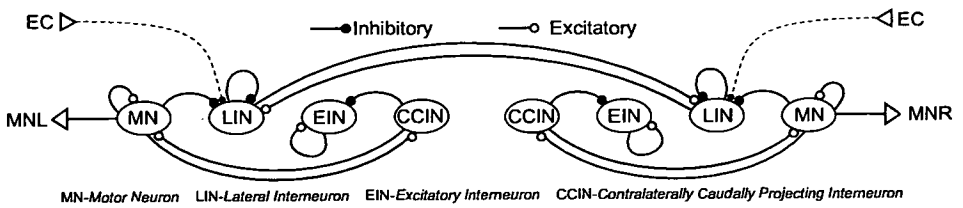


Fig. 2. Evolved controller for the system in fig 1. A fully connected topology was developed. However, for clarity, only weights> 1.7× average synaptic weight are illustrated.

It is proposed that we may be able to apply the lamprey CPG to such an articulated WEC. However in order to test this hypothesis, we must start with a simple system. If we consider that we can approximately model an articulated WEC as a series of point absorbers limited to heave only, whereby the power out-take (C_{PTO}) is between adjacent floats, we can further simplify this and consider the control of a single float as in fig. 1.

This is convenient as it has been shown [2] that a single neural CPG segment, see fig. 2, will oscillate at a natural frequency defined by the weights of the network, modulated by the sensory input. In future work we will show the extension of this to include interconnected buoys as described above and how we can apply a series of interconnected CPG segments as the control.

3.1 Developing a Neural Controller

The mechanical simulation described in section 2 was coupled to the neural network using the buoy velocity ($\dot{z}$) as a sensory feedback input. This allowed the period of oscillation of the CPG network to adapt to that of the WEC device. Furthermore, optimisation of the network weights allows the adjustment of the phase difference range and the bandwidth over which the CPG output *(the neural network output)* and the device displacement could become matched. It was found that an effective latching strategy could be implemented by fixing the buoy displacement at extremity by using $C_{PTO} \ggg optimum$ when $\dot{z} = 0$ [6] and using the CPG output to trigger the release, where $C_{PTO} = optimum$.

The synaptic weights of the network were optimised by a genetic search process. The fitness of each individual was rated using the average power developed over four separate 60 second simulations, each at a different, constant wave period. The Genetic Algorithm (GA) was also given the freedom to evolve the value of $C_{PTO\ (optimum)}$ as it was not clear exactly what value would be appropriate to the evolved solution.

Using a real valued population and taking advantage of the network symmetry, individual chromosomes consisted of 73 variables each of 6dp precision, giving a search space of $10^{73\times6}$. For the results here, three separate evolutions were invoked, each of which converged within 500 generations. The average variation in individual weights between these solutions was less than 2% and the

"fitnesses" of the final solutions matched to within 0.2%. The evolved result is shown in fig. 2.

4 Results

The results in fig. 3 show that the evolved strategy was successful over a useful range of frequencies, but it is interesting to note that it was not possible to produce a network that would provide latching control close to the resonant period of 2.9s. This can be explained relatively straightforwardly. If T_L defines the duration of latching, then the phase lag introduced by latching can be described as: $\phi_i = \frac{T_L}{Period}\pi$. For maximum power transfer, the optimal phase difference between the wave and buoy displacement is $\phi = \phi_n + \phi_i = \frac{\pi}{2}$. At resonance however, the natural phase shift of the buoy is $\phi_n = \frac{\pi}{2}$, so the induced additional phase ϕ_i (though latching) should be 0 *at resonance.* Close to resonance however, a lower limit is imposed upon T_L by the neuron model, forcing $\phi > \frac{\pi}{2}$, resulting in sub-optimal power transfer. Under these conditions, since the network phase locks to the buoy displacement rather than to the water level, a feed-forward effect reinforces this error and neural control fails. In theory it would be possible to adjust the neuron coefficients to reduce the minimum value of T_L, ultimately side-stepping the control failure at resonance. However, as the release point must occur *after* the fix point, then due to the discrete nature of the simulation T_L can never equal zero as it is limited by the minimum time-step of the simulation.

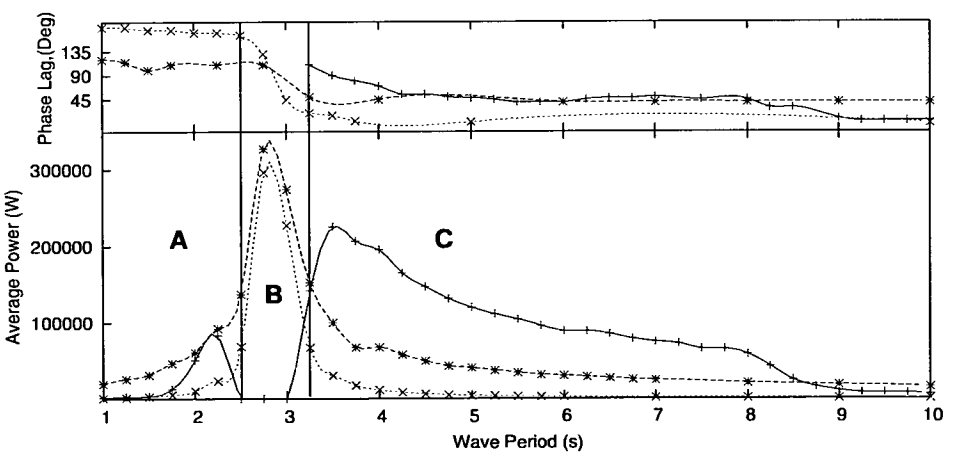


Fig. 3. Phase and power response for regular waves with periods between 1 and 10 seconds. Within region "C" the neural controlled latching strategy (+), can be seen to outperform optimal real damping (∗) and the uncontrolled system (×) over much of this frequency range. Region "B" illustrates areas where a solution could not be evaluated. Results in region "A" are not of interest in this paper.

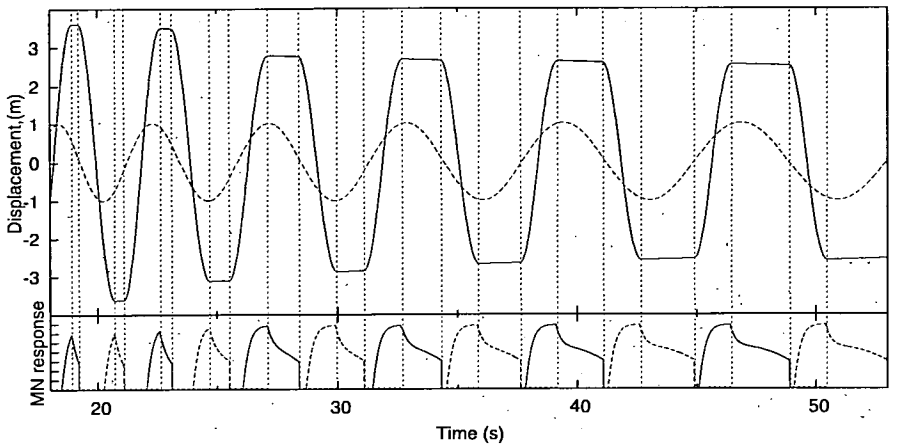


Fig. 4. Response of the system to a changing period from 3 to 10s over a duration of 35s. The top pane shows the wave displacement (dashed) relative to the device displacement (solid), while the lower pane shows the Motor Neuron (MN) output from the network. The vertical lines indicate the latch points for the system. Although these are not conditions experienced in reality, temporal scales for areas of interest are correct.

In order to concentrate on producing a general solution for periods above 3.5s, the network was evolved using four regular wave periods of 3.5, 4, 5.5 & 7 seconds with overall developed power used as the fitness. Over this bandwidth it can be seen to show a significant improvement over optimal real damping.

The GA evolved a single value of C_{PTO} for all periods. However in optimal real damping an ideal value was computed for C_{PTO} at each period and this value increased significantly with the period. From fig. 3 it can be seen that there is a distinct reduction in power from the neural control above its evolved bandwidth (>8s). It was seen that this reduction was due to the value of C_{PTO} being too low at these longer periods. By simply increasing this value for periods 8s< *period* <10s, increases of up to 246% were observed.

The ability of the neural controller to adjust to changing input conditions is clearly illustrated in fig. 4. The sensory input to the network allows the oscillatory period of the network to adapt to the displacement of the buoy and hence to define the latching release points accurately. Even though a practical strategy is implemented, it is possible to see that the ideal $\frac{\pi}{2}$ phase shift is not obtained. This is because future knowledge of the wave is *required* for this optimum condition. However, by using the neural system as we have, we can effectively extrapolate the correct release point from the past knowledge contained within the network. This can be seen in fig. 4 whereby the release point is somewhat too early. Nevertheless, good performance is maintained over an input wave sweeping from 3.5 to 10s.

5 Concluding Remarks

We have shown that "causal" latching control for wave energy converters (WEC), based on a model of a biological (CPG) controller, can be developed using genetic search techniques. This method produces controllers that can outperform optimised real damping over a wide range of sea conditions, with an idiosyncrasy at resonance that can be avoided straightforwardly. Although the wave conditions demonstrated here are far from real conditions, the controller demonstrates a basic ability to adapt accurately to changing input conditions. Extension to testing in realistic-wave conditions will be covered in a future paper.

This work is based upon a simple abstraction of a real WEC device that captures much of the richness of a more complex model and we are optimistic that the promising results reported here will translate into a simple, implementable strategy for control of power out-take in real WEC devices. Furthermore, we speculate that, using naturally-evolved examples of "neural" computing and control structures as a starting point, it may be possible to evolve novel solutions to further hard problems by altering the constraints under which the neural solution is optimised.

References

1. K. Budal and J. Falnes. Interacting point absorbers with controlled motion. In *Power from Sea Waves*, pages 129–142, Edinburgh, UK, 1979.
2. O. Ekeberg. A combined neuronal and mechanical model of fish swimming. *Biological Cybernetics*, pages 363–374, 1993.
3. J. Falnes. *Ocean Waves and Oscillating Systems: linear interactions including wave-energy extraction.* Cambridge University Press, 2002.
4. J. Falnes. Optimum control of oscillation of wave-energy converters. *International Journal of Offshore and Polar Engineering*, 12(2), June 2002.
5. J. Falnes and K. Budal. Wave power conversion by point absorbers. *Norwegian Maritime Research*, (4), 1978.
6. M. J. French. A generalized view of resonant energy transfer. *Journal Mechanical Engineering Science*, 21(4):299–300, 1979.
7. S. Grillner, P. Wallén, and L. Brodin. Neuronal network generating locomotor behaviour in lamprey. *Annu. Rev. Neurosci*, pages 169–199, 1991.
8. R. E. Hoskin and N. K. Nichols. Latching control of a point absorber. In *3rd International Symposium of Wave, Tidal, OTEC and small scale Hydro Energy*, pages 317–329, Brighton, UK, May 1986.
9. P. Wallén and S. Grillner. Central patern generators and their interaction with sensory feedback. In *Proceedings of the American Control Conference*, volume 5, pages 2851–2855, 1997.
10. www.oceanpd.com. Ocean power delivery. Internet Site.