Model Predictive Control of a Wave Energy Converter: Archimedes Wave Swing

Abstract:

Wave energy is a promising source of clean and renewable energy. In order to tap this energy various designs for wave-energy converters are currently under development. One of these devices is the Archimedes Wave Swing. It is basically a submerged air vessel consisting of a floater and a stator. The floater is free to move vertically under varying wave pressure, while the stator is anchored to the sea bed. Energy is extracted from the relative motion of the two parts.

The control objective is to optimize the power produced while ensuring that the motion of the floater remains within certain prescribed limits. The main difficulty is due to the high degree of irregularity of ocean waves. Two consecutive waves can have significantly different heights and periods, and the controller should be able to deal with that.

The controllers are designed based on two different principles. The first method controls the trajectory so that the velocity of the floater remains in phase with the excitation force. Model predictive control is used in order to be able to eact to the constraints ahead of time. The second method makes use of a prediction of the excitation force in order to calculate the control force which maximizes the energy produced.

Paul Gieske
1020897

Contents

Chapter 1

Introduction

A rapidly increasing oil price and increased awareness of global climate change have led to a search for clean and renewable energy. One option is harnessing the energy in ocean waves. A number of wave energy converters (WECs) are currently being developed and have reached the precommercial stage of development. One of these is the Archimedes Wave Swing (AWS). The AWS is a cylindrical underwater air-reservoir consisting of a floater which is free to move vertically, and a stator which is anchored to the sea bed. The position of the floater varies with the changing hydrodynamic pressure and energy is extracted from the motion.

The AWS is subject to some constraints: the maximum floater position and velocity are constrained and the generator force is limited to 1 MN. Furthermore, ocean waves are highly irregular, meaning that there is a large variation in the wave periods and wave heights for the same sea conditions. The motion of the floater must be controlled in order to optimize the wave energy converted while satisfying the given constraints.

The combination of the irregularity of the waves and the importance of the constraints obedience makes a predictive control strategy a suitable candidate. Predictive control can react in advance to the incoming waves, allowing it to safely operate closer to the constraints. For example, the controller provides an extra braking force only for particularly large waves where the constraints are endangered.

The report is structured as follows. Chapter 2 describes the control problem in more detail. Chapter 3 describes the modelling done to design and evaluate the controllers. It includes models of the system as well as models used to estimate and predict the wave force. Chapter 4 describes the controller design. Chapter 5 evaluates the controllers. Finally, chapters 6 and 7 contain the conclusions and recomendations respectively.

Chapter 2

Control Problem Description

This chapter briefly describes the control problem. First the AWS is introduced in section 2.1. A more complete overview of the system is given in chapter 2 of the literature study report. Section 2.2 briefly describes the wave profiles used to generate the results presented in this report. Next the basics of the control problem are described in section 2.3. Finally the control strategy to be used is introduced in section 2.4.

2.1 The AWS Design

Figure 2.1 shows the working principle of the AWS. The AWS is a submerged air-vessel consisting of two parts. The top part (the floater) is free to move vertically while the bottom part (the stator) is anchored to the sea bed. When the trough of the wave passes over the AWS the water pressure acting on it is especially low and the air inside expands, pushing the floater upwards. When the peak of the wave passes over the AWS the water pressure acting on it is especially high, pushing the floater downwards. A generator extracts energy from the relative motion of the two parts.

Figure 2.2 shows a simple schematic of the AWS. The motion of the AWS is constrained. If the vertical displacement floater exceeds a certain limit a set of water brakes will activate providing an extra damping force. Additionally, there are rubber end stops at the furthest allowable limit. The velocity of the floater is also be limited because large velocities induce large electrical currents.

The design also incorporates a pump which can pump water into and out of the AWS, changing the air pressure. This is done in order to tune the natural frequency of the AWS to match it with the average excitation period at that time.

The generator is an experimental permanent magnet linear generator. The main difference to a conventional generator is that the motion of the rotor is linear. The generator provides a damping force to the floater when

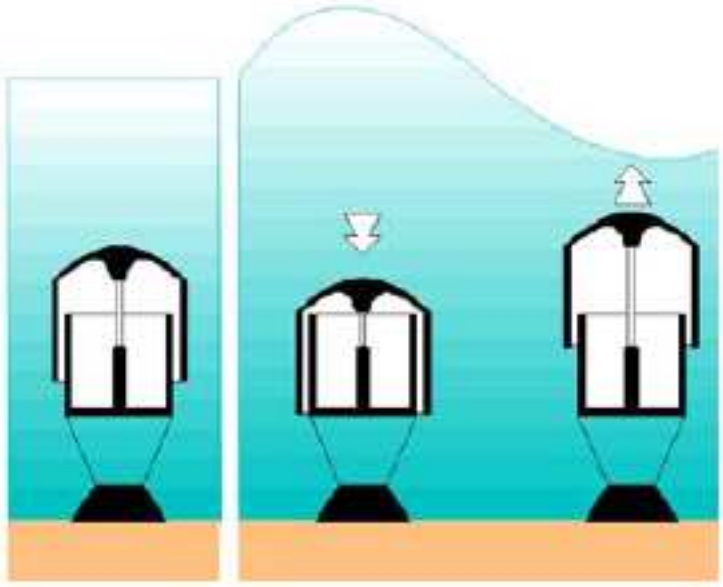

Figure 2.1: Working Principle of the AWS

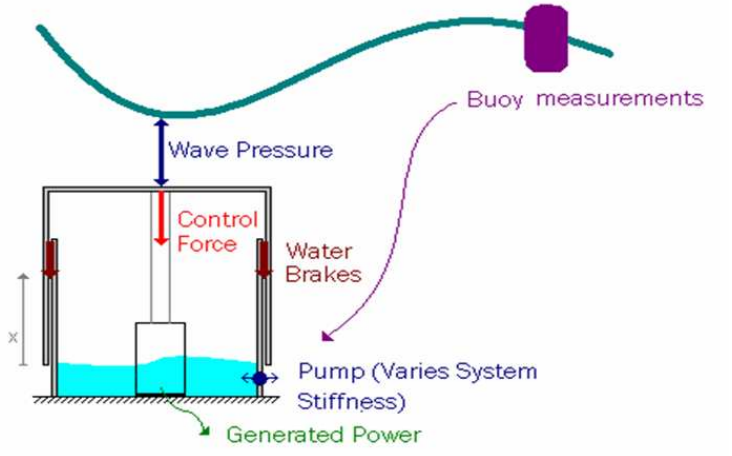

Figure 2.2: Schematic of the AWS

it extracts energy from the motion. It is also capable extracting energy from the electrical grid to providing a control force in the same direction as the velocity.

There may be a floating buoy present in the neighbourhood of the device. If so measurements of the surface waves are also available to the controller.

In 2004 a prototype of the AWS was implemented off of the coast of Portugal. The floater has a diameter of 9.5 m and weighs $0.4{\cdot}10^6$ kg. The total height with the floater at mid position is 38 m, with an air volume of 3000 m^3. A total of 1500 m^3 of water can be pumped into the AWS allowing it to be tuned to between 7 and 13 second wave periods.

2.2 Ocean waves

Ocean waves are irregular on the short term but vary on the long term as well depending on weather conditions. The average characteristics of the waves over intervals of 30 minutes is refered to as the sea state.

In this report sea states are characterized by the significant wave height (H_s) and the average upwards zero-crossing period (T_z). The significant wave height is defined as the average of the one third highest waves. The average upwards zero-crossing period is defined as the average interval between upwards crossings of the mean water level.

This report makes use of synthetically generated waves. The wave profiles are generated by filtering white noise such that the resulting signal has the desired power spectrum. The desired power spectrum is based on the JONSWAP power spectrum. The spectrum is modified to resemble the spectra attained from the measurements at the test site. The measurements were taken at the AWS prototype test site, off of the coast of Portugal. The JONSWAP spectrum is given as:

$$J(H,T,\gamma,\omega) = \frac{5.23\pi^3 H_s^2}{T^4\omega^5}\exp\left[-32.39\frac{\pi^3}{T^4\omega^4}\right]\cdot\gamma^p, \tag{2.1}$$

where

$$\begin{aligned} p &= \exp\left(-\left(\frac{0.191\omega T-1}{\sqrt{2\sigma}}\right)2\right), \\ \sigma &= 0.07\ for\ \omega \leq 5.24/T \\ \sigma &= 0.09\ for\ \omega > 5.24/T \end{aligned} \tag{2.2}$$

The following spectrum is used to generate the wave data:

$$J_{mod} = 4J(H, 0.9T_z, 2.8, \omega) \tag{2.3}$$

Chapter 3.1 of the literature study discusses ocean waves more deeply. References [11] and [12] and chapter 3.1.4 of the literature study report describe the procedure for generating the wave profiles.

Parameter	Value
c_1	250 kW
c_2	20 kW
c_3	10 kW
F_{rated}	1 MN
v_{rated}	2.2 m/s

Table 2.1: Generator parameters

2.3 Control Problem

The AWS prototype made use to an experimental linear generator. The generator is capable of producing a 1 MN control force (F_{gen}) parallel to the floater motion. The generator force is a result of extracting energy from the motion and can also be used to control the motion of the floater.

The power produced by the generator is given by the following equation:

$$P = -F_{gen}\dot{z} - c_1\left(\frac{F_{gen}}{F_{rated}}\right)^2 - c_2\left|\frac{\dot{z}}{v_{rated}}\right| - c_3 \tag{2.4}$$

The first term expresses the power extracted by the generator from the motion. Note that when the control force and the velocity are the same sign, the power *produced* is negative, i.e. it costs energy to increase the speed of the floater. The second term represents the resistive losses, also known as copper losses, and is approximated as proportional to the square of the generator force. The third term represents the iron losses and is approximated as proportional to the speed of the floater. The final term represents the power used by the AWS for miscellaneous functions such as, for example, cooling the converter. Table 2.3 lists the values for the parameter given in equation 2.4. Reference [4] gives a description of the generator.

The control objective is to maximize the energy produced while satisfying the constraints on the motion as well as the limitations of the control force. Note that maximizing the energy output is not equivalent to maximizing the power produced at every point in time. The generator can extract power from the grid to facilitate the motion of the AWS, thereby increasing the overall amount of energy produced.

The control problem is made more challenging by the nature of ocean waves. The surface waves are highly irregular, meaning that there is a significant difference between the periods and heights of consecutive waves, which the system must be able to handle. References [2], [3], [8], [9] and [10] discuss various manners of using control to deal with the wave irregularity.

2.4 Control Strategy

The significance of the constraints in combination with the irregularity of the excitation force make model predictive control a promising control methodology. For example, if an unexpectedly large wave approaches the AWS, endangering the constraints, a predictive controller can react to it in advance, thereby allowing it to safely operate close to the constraints. An example of a model predictive control technique used to handle irregular waves is discussed in reference [3], where a prediction is used to calculate the optimal latching relase time.

Basically, two kinds of controllers are designed:

- Trajectory control: A controller of this kind controls the velocity of the floater to ensure that it is in phase with the excitation force. The magnitude is defined as a constant scaling of the excitation force. The scaling factor is based on an equation for the optimal velocity trajectory, as derived in chapter 4.3.2 of the literature study report and chapter 6 of *Ocean Waves and Oscillating Systems* [6]. The expression is only valid when the generator losses and constraints are neglected and the motion is unconstrained. The true optimal trajectory is very difficult to calculate.

- Energy maximization: An alternative is to design a controller which attempts to optimize the energy produced directly. The difficulty with this kind of controller is in formulating an optimization problem which accurately describes the energy produced as a function of the predicted excitation force, the initial state and the future control force.

In addition, two other possibilities are suggested. The possibility of switching between the reference tracking and energy maximizing controllers is discussed. Also a method for expressing the electrical energy produced by the generator approximately in order to allow the optimization problem to be solved with less calculations is discussed.

In order to design the controllers it is necessary to have an estimate of the future excitation force. The excitation force can be predicted by extrapolating it from past values. This is be done using an autoregressive (AR) model, as is discussed in section 3.3.

To predict the excitation force past values are needed. These can be provided by use of a pressure sensor. Including a pressure sensor however, would influence the overall cost and reliability of the system. In case there excitation force cannot be measured, the excitation force can be estimated from the motion of the AWS. This is discussed in section 3.4.

In order to investigate the effects of the prediction and estimation on the performance, the code is written to be able to handle the following three cases:

- The excitation force is known a priori. This is an ideal case. Unless otherwise stated the results presented in this report make use of knowledge of the future excitation force.
- The excitation force is known for past and present, and the future excitation force must be predicted. In this case it is assumed that the AWS is equipped with an ideal pressure sensor.
- The excitation force is not known at all. It must first be estimated and then extrapolated.

By comparing the performance for each of the above cases the need for improved estimation or prediction can be evaluated.

Chapter 3

Modelling

This chapter discusses the models used to design and evaluate the controller. Section 3.1 describes the non-linear model of the AWS. This model is used to evaluate the closed loop performance, and to estimate a linear model. Section 3.2 describes the linearization of the model. Section 3.4 describes a model which is used to estimate the excitation force. Finally, section 3.3 describes the model used to make a prediction of the excitation force.

3.1 AWS model

The AWS is a system with one degree of freedom: the floater is free to move vertically only. The motion of the floater is thus determined by the sum of the vertical forces. By Newton's second law the acceleration of the floater is equal to the sum of the forces divided by the mass of the floater.

$$m_f \ddot{z} = F_{bear} + F_{drag} + F_{grav} + F_{hs} + F_{rad} + F_{sp} + F_{wb} + F_{gen} + F_{end} + F_e \quad (3.1)$$

The forces are each described below including the equations used to model them. Chapter 2.3 of the literature study report and reference [4] give more information. Table 3.1 gives the values for the parameters used. Figure 3.1 shows a schematic illustrating the dimensions involved. Appendix B.1 contains the code used to implement the model.

F_{bear} is the force due to friction in the bearings. It depends on the normal force felt by the top (F_{Ntop}) and bottom (F_{Nbot}) bearings, and the bearing friction coefficient (μ).

$$F_{bear} = -\mu \cdot \text{sign}(\dot{z}) \cdot (|F_{Ntop}| + |F_{Nbot}|) \quad (3.2)$$

F_{Ntop} and F_{Nbot} depend on the position of the bearings and the horizontal load profile of the floater. The horizontal load profile can be calculated

Symbol	Description	Value
C_{DDW}	drag coefficient for downwards velocities	0.4
C_{DUP}	drag coefficient for upwards velocities	0.2
d_{bot_0}	distance from floater bottom to bottom bearing at mid-position	10 m
d_{top_0}	distance from floater bottom to top bearing at mid-position	19 m
d_f	depth to the floater top at mid-position	11 m
g	acceleration due to gravity	$9.8\ \mathrm{m/s^2}$
h_f	floater height	28.5 m
m_{add}	added mass	$3.55 \cdot 10^5$ kg
m_f	floater mass	$4 \cdot 10^5$ kg
p_{amb}	ambient pressure	$1 \cdot 10^5\ \mathrm{N/m^2}$
S_f	inner cross-sectional area of the floater	$79\ \mathrm{m^2}$
S_F	outer cross-sectional area of the floater	$95\ \mathrm{m^2}$
β_{wb}	water brake coefficient	$1.5 \cdot 10^6\ \mathrm{kg/m}$
γ	heat capacity ratio	1.4
θ	end stop position	4.5 m
κ	water brake limit	4 m
μ	bearing friction coefficient	0.1
η_T	tide level	0 m
ρ	water density	$1.025 \cdot 10^3\ \mathrm{kg/m^3}$

Table 3.1: Simulation parameters

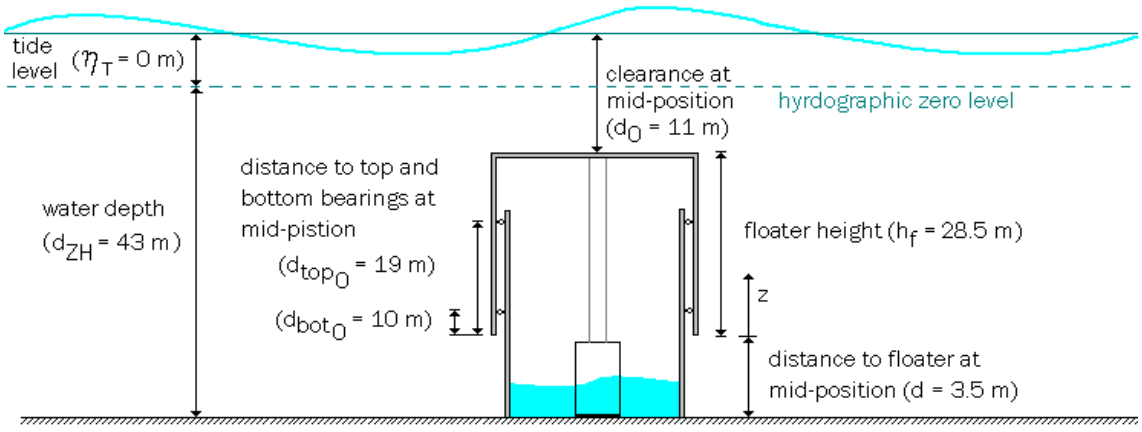

Figure 3.1: AWS dimensions

from the sea surface elevation using linear wave theory. For an overview of the linear wave theory used, see chapter 3.2 of the literature study report and reference [6]. The hydrodynamic loads are calculated as the resulting horizontal force (F_x) and horizontal moment (M_x) relative to the bottom of the floater at midposition. Appendix B.3 gives the code used to calculate the horizontal loads. The normal forces acting on the bearings can then be calculated using classical mechanics.

F_{drag} is the drag force of the water acting on the floater, and is expressed in terms of the upwards and downwards drag coefficients (C_{DUP} and C_{DDW}), the outer cross-sectional area of the floater (S_F), the water density (ρ) and the floater velocity ($\dot{z}$):

$$F_{drag} = \begin{cases} -\frac{1}{2}\rho S_F \dot{z}|\dot{z}|C_{DUP}, & \dot{z} \geq 0 \\ \\ +\frac{1}{2}\rho S_F \dot{z}|\dot{z}|C_{DDW}, & \dot{z} < 0 \end{cases} \tag{3.3}$$

F_{gen} is the force applied by the generator and can be considered the control input. The force is used to control the motion and to extract energy. The power produced by the generator is given as the product of the velocity and the generator force minus the generator losses. The losses are given in terms of 3 constants (c_1, c_2 and c_3), the designed maximum generator force (F_{rated}) and the designed maximum floater velocity (v_{rated}).

$$P = -F_{gen} \cdot \dot{z} - c_1 \left(\frac{F_{gen}}{F_{rated}} \right)^2 - c_2 \left| \frac{\dot{z}}{v_{rated}} \right| - c_3 \tag{3.4}$$

F_{grav} is the weight of the floater, and is given as the product of the floater mass (m_f) and the acceleration due to gravity (g).

$$F_{grav} = -m_f g \tag{3.5}$$

F_{hs} is the force due to the hydrostatic pressure acting on the AWS. It varies with the depth (d_f), tide-level (η_T) and the floater position (z). It is parameterized by the outer (S_F) and inner (S_f) floater cross-sectional areas, the floater height (h_f), the water density (ρ) and the ambient pressure (p_{amb}).

$$\begin{aligned} F_{hs}(z) = -S_F(\rho g(d_f + \eta_T - z) + p_{amb}) + \\ + (S_F - S_f)(\rho g(d_f + \eta_T + h_f - z) + p_{amb} \end{aligned} \tag{3.6}$$

F_{rad} is a force acting on the AWS which is the reaction of the wave being radiated from it. It consists of an extra inertial term (m_{add}) and a convolution integral representing the memory of the fluid. Chapter 3.2.4 of the literature study report and reference [6] cover the derivation. The convolution integral in 3.7 can be approximated in a number of ways as

shown in appendix C. For the results presented in this report, the memory term was approximated as the radiation force being modelled as a constant damping, where the damping value depends on the average zero-crossing wave period.

$$F_{rad} = -m_{add}\ddot{z} - \int_0^t R(t-\tau)\dot{z}(\tau)d\tau \tag{3.7}$$

F_{sp} is the force due to the gas pressure inside the AWS. It is expressed as a function of the floater position (z) in terms of the equilibrium spring force ($\bar{F}_{sp}$), the equilibrium position ($\bar{z}$), the heat capacity ratio (γ) and the equivalent spring length at equilibrium ($\bar{L}_G$). The value for equivalent spring length is set to the desired value by pumping water into or out of the AWS, in order to tune or detune the AWS to the excitation force. The AWS can be tuned to a natural period of between 7 and 13 seconds. The results presented in this report are for a tuning period of $1.3{\cdot}T_z$, but limited to between 7 and 13 seconds.

$$F_{sp} = \bar{F}_{sp}\left(1 + \frac{z-\bar{z}}{\bar{L}_G}\right)^{-\gamma} \tag{3.8}$$

F_{wb} is the force applied by the water brakes. The water brakes act as an extra damping force when a limit in the vertical displacement is exceeded:

$$F_{wb} = \begin{cases} 0 & z < \kappa \\ \\ -\beta_{wb}\dot{z}^2, & z \geq \kappa \end{cases} \tag{3.9}$$

where κ is the water brake limit and β_{wb} is the water brake coefficient.

F_{end} is the reaction force of the rubber end stops in case they are hit. In principal this should never occur but they are nevertheless included in the model to prevent unrealistic results for simulation case where the constraints are broken. The end stop reaction force is modelled as a force which attempts to halt the floater within 0.1 seconds:

$$F_{end} = \begin{cases} 0, & z < \theta \\ (m_f + m_{add})\dot{z}/0.1, & z \geq \theta \end{cases} \tag{3.10}$$

where θ is the end stop position.

F_e is the excitation force, also refered to as the vertical hydrodynamic force. The excitation force is a function of the wave profile. Chapter 3.2 of the literature studyand reference [6] describe the linear wave theory used to calculate the excitation from the sea surface elevation:

$$F_e(d,\omega) = \rho g S_F \eta(\omega) K_P(d,h,\omega) \tag{3.11}$$

where d is the depth, η is the surface elevation and K_P is the decay factor. K_P is a function of the depth to the sea bed (h), the depth and frequency.

$$K_P(\omega,h,d) = \frac{\cosh(k(\omega)(h-d))}{\cosh(k(\omega)h)} \tag{3.12}$$

where $k(\omega)$ is the wave number. The excitation force is calculated for a floater at mid-position. This is an approximation. See section 5.4.2 for a discussion on the approximation. Appendix B.3 gives the m-code used to translate the surface elevation to the excitation force.

The excitation force and the force of gravity are external forces since they do not depend on the motion of the system. The coulomb friction in the bearings depends both on the position of the floater as well as the resulting horizontal force acting on the floater and hence depends on both external and internal signals.

3.2 Linearizing the AWS model

Making use of a linear model to predict the response of the AWS improves the speed of solving the optimization problem, and also allows the problem to be formulated as a quadratic problem when the cost function is expressed as a linear function of the state.

The system can be seen as a non-linear mass-spring damper, whereby the hydrodynamic forces caused by the waves are seen as the external inputs. The non-linear system can hence be approximated as a second order linear system parameterized by a stiffness coefficient (k), a damping coefficient (d), and a value for the inertia felt by the AWS (m):

$$\begin{pmatrix} \dot{z} \\ \ddot{z} \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ -\frac{k}{m} & -\frac{d}{m} \end{pmatrix} \begin{pmatrix} z \\ \dot{z} \end{pmatrix} + \begin{pmatrix} 0 \\ \frac{1}{m} \end{pmatrix} F_e + \begin{pmatrix} 0 \\ \frac{1}{m} \end{pmatrix} F_{gen} \quad (3.13)$$

The inertia felt by the device is the sum of the mass of the floater (m_f) and the extra intertia felt by the floater due to it being submerged, referred to as the added mass (m_{add}).

$$m = m_f + m_{add} \quad (3.14)$$

The floater mass is known and the added mass can be determined theoretically or experimentally.

An appropriate value for the stiffness coefficient is the stiffness at the equilibrium position. The equilibrium stiffness is actively tuned so that the AWS is in resonance with the excitation force. The same stiffness value to which the AWS is tuned is used in the linear model.

$$k = \frac{\gamma g \bar{F}_{sp}}{\bar{L}_G} - \rho g S_f \quad (3.15)$$

The damping force is not easy to linearize. There are three damping forces present.

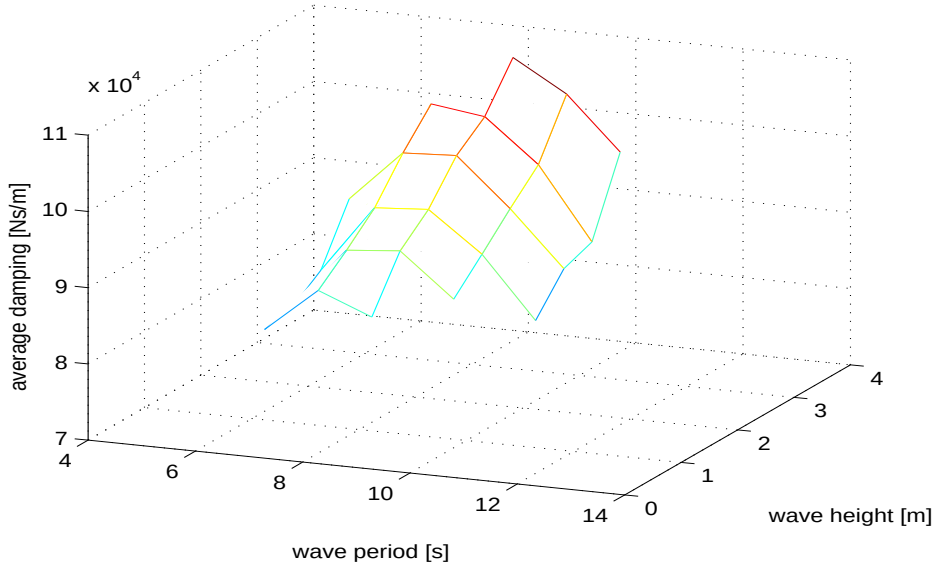

Figure 3.2: Average damping coefficients for a range of sea-states

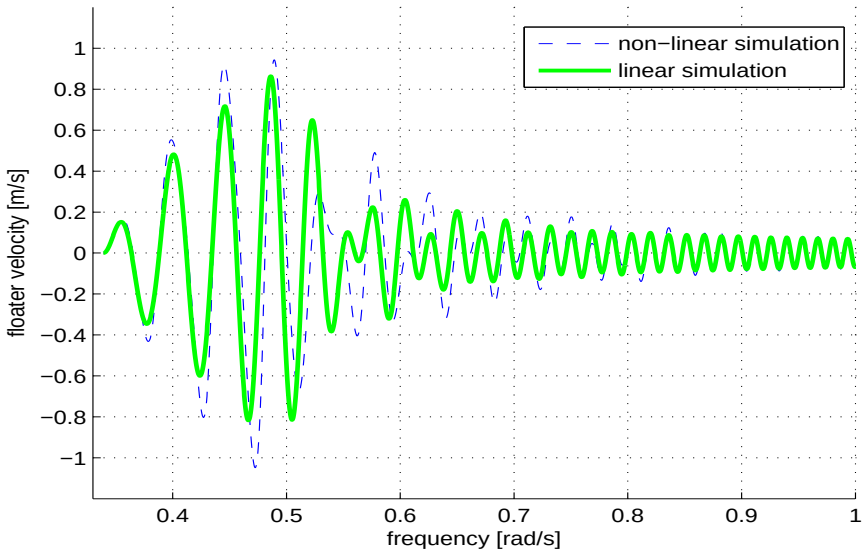

Figure 3.3: Open loop response to a chirp signal

- The drag force is strongly non-linear. It varies proportionally to the square of the velocity, and the drag coefficient changes with the sign of the floater velocity. When the sea state is rough the root mean square of the drag force will be larger than when the sea state is calm.

- The radiation force depends on the frequency of the floater velocity, and hence depends on the wave period. The radiation force is highest for a period of between 7 and 8 seconds.

- The magnitude of the friction in the bearings depends on the position of the floater and the magnitude of the horizontal hydrodynamic loads acting on it. When the waves are higher, the horizontal forces are larger and it is expected that the average coulomb damping at the bearings will be higher. Only the direction of the friction is related to the velocity.

In order to model the damping forces linearly an equivalent linearization is performed. The idea is to get an average value for 'best-fit' damping coefficients as a function of the sea-state.

The AWS motion is simulated for 1000 real sea states using the most detailed model described in section 3.1 whereby a control force opposing the excitation force of a magnitude of 0.8 times the excitation force is assumed. The sea states used are taken from sea surface measurements taken at the test site of the AWS prototype. This excites the motion of the floater to the expected amount for each sea state. The average linear damping coefficient is calculated for each sea state as $d = |\bar{F}_d|/|\bar{\dot{z}}|$ where $|\bar{F}_d|$ is the time average of the sum of the absolute values of all the damping forces and $|\bar{\dot{z}}|$ is the time average of the absolute values of the floater velocity. Figure 3.2 shows the average damping coefficients attained for different sea states. The results used in this report make use of the linear dmping coefficient $d(H_s) = 5200 \cdot H_s + 87000$.

Determining the linear damping coefficient in this way has the advantage that it finds an average value over the entire operating range. Analytically linearizing the model about a certain point would be representative only in the neighbourhood of that point. For example, consider linearizing the model around $\dot{z} = 0$. The damping coefficient due to the drag force is equal to zero because the drag is proportional to the square of the velocity. The damping coefficient due to the bearing friction is equal to zero because the magnitude of the force is not related to the velocity. This leaves only the damping caused by the reaction of the radiated wave which badly mis-represents the AWS for non-zero floater velocities.

Figure 3.3 compares the open loop responses of the linear and non-linear models for a chirp input signal. The input signal is a sine function with a frequency which varies from 0.34 rad/s to 2 rad/s and has a magnitude of $1{\cdot}10^5$ N. The tuning frequency of the AWS is set to 0.5 rad/s.

The damping in the linear model is set to $96.1 \cdot 10^3$ Ns/m corresponding to a significant wave height of 1.75 m. The friction in the bearings for the non-linear model is modelled using the horizontal loads as calculated for a sea state with significant wave height of 1.75 m and a wave period of 9.5 seconds. The responses of the models appear to be quite similar and it is hopeful that the linear model is sufficient to be used by the model predictive controller.

3.3 Predicting the Excitation Force

The excitation force is highly irregular and directly influences the energy input to the system. In order to optimize the controller a prediction of the excitation force is required.

The excitation force is predicted by means of an auto-regressive (AR) model. For a specific sea state an AR model can be identified given enough values of the measured or estimated excitation force. This model can then be used to make short-term predictions. The model should be re-identified whenever the sea state changes significantly. Auto-regressive models are of the form:

$$y[n+1|n] = a_1 y[n] + a_2 y[n-1] + a_3 y[n-2] + \cdots + a_{N-1} y[n-N+2] \quad (3.16)$$

where N is the order of the model and a_1 up to a_{N-1} are constant coefficients.

The wave prediction is implemented by the controller as follows. First the AR model is identified using the matlab function *ar.m*. This is only done once for a given sea state. The AR model is identified based on a time series of 500 values with a step size equal to the controller step size. The controller step size is typically between 0.3 and 0.6 seconds depending on the tuning period (the step size is chosen such that there are 30 steps within one tuning period). The order of the model is set to 40.

Next a prediction of the excitation force is made using the 40 previous values for the excitation force. This prediction can be used by the model predictive controller.

To ensure a reasonable excitation force model it is important that the step size should be much smaller than the average period of the excitation force. It is also important that the identification data spans a long enough time period to ensure that the identified model is stable.

Figure 3.4 gives a typical example of the prediction. As can be seen, in this case, the prediction is quite accurate on the short term but starts to deviate from the true excitation force after some time. In general the prediction method can only be used for short term predictions of less than one period. Appendix B.1.3 contains the code used to identify the excitation force prediction model.

3.4 Estimating the Excitation Force

In order to predict the excitation force by the method described above, past values for the excitation force must be given. It is possible to use a pressure sensor to measure this force although this would decrease the reliability of the design. Alternatively, it is possible to estimate the force from the motion of the AWS.

It is assumed that the measured vertical displacement signal is of sufficiently high quality so that it is possible to calculate the acceleration. Section 5.4 discusses the case where the acceleration cannot be estimated in this manner. From Newtonian mechanics:

$$F_e = m\frac{\partial^2 z}{\partial t^2} - F_d - F_k - F_{wb} - F_{gen} \tag{3.17}$$

The damping force (F_d) is estimated using the average linear damping attained in section 3.2. The spring force (F_k) is attained using the same equations as those used in the non-linear model described in section 3.1. This is an ideal case since the exact stiffness force would not be known in real life. Section 5.4 demonstrates that using an approximate model of the spring stiffness is also sufficient to estimate the excitation force. The water brake force (F_{wb}) and the generator force (F_{gen}) are known to the controller.

The attained estimate is filtered with a band pass filter. High frequencies introduced by sudden changes in the generator force and water brakes, and those introduced by the erratic coulomb friction in the bearings, are filtered out. Low frequencies due to numerical errors which introduce drift are also filtered out. The low pass and high pass filters used are given as:

$$LP(s) = \frac{400}{(s+20)^2} \tag{3.18}$$

$$HP(s) = \frac{s^2}{(s+0.1)^2} \tag{3.19}$$

Figure 3.5 gives the bode plot of the band pass filter. Frequencies lower than 0.2 rad/s and higher than 10 rad/s are filtered out. The frequency of the excitation force is always between 0.3 rad/s and 1.6 rad/s depending on the sea state. The filter is hence not expected to influence the magnitude of the estimate, but may cause a small phase shift for low frequencies.

Figure 3.6 shows a typical result for two sea states with a zero-crossing wave period of 5.5 seconds and 11.5 seconds, each with a significant wave height of 2.25 m. The estimate is in general reasonable, but shows some error due to the damping estimate.

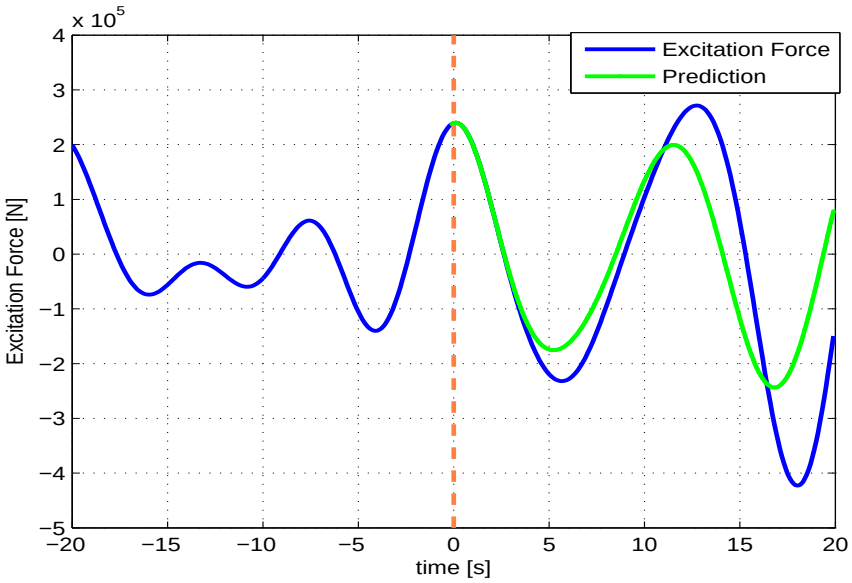

Figure 3.4: Prediction of the Excitation Force

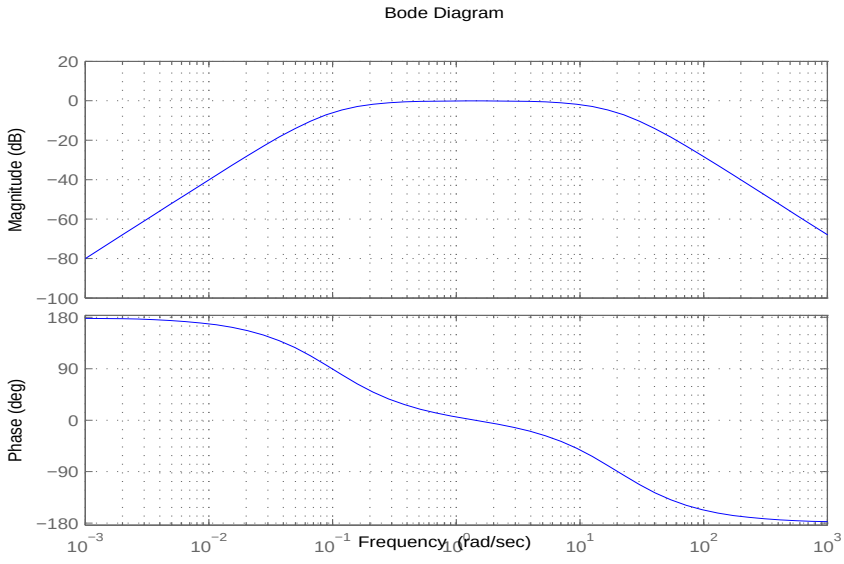

Figure 3.5: Band pass filter used to estimate the excitation force

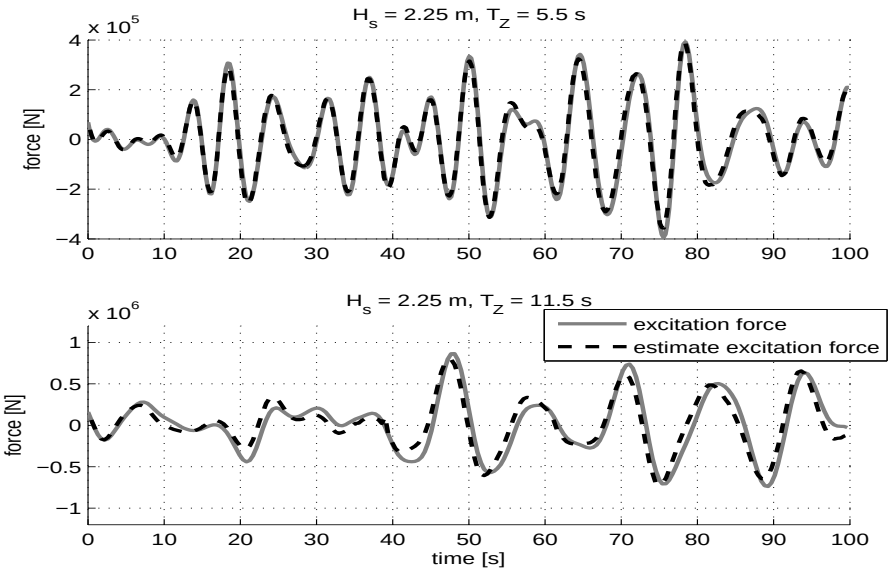

Figure 3.6: Excitation force estimate

Chapter 4

Controller Design

This chapter discusses controller design. Basically two types of controllers have been designed. One controls the trajectory of the device, as discussed in section 4.1. The other attempts to optimize the energy produced, and is discussed in section 4.2. Additionally two alternative possibilities are suggested. The possibility to switch between the two controllers is discussed in section 4.3. Finally, attempts to approximate the expression for the energy produced in order to lower the computational requirements of the energy maximizing controller are discussed in section 4.4.

4.1 Reference Tracking for Phase Control

Phase control is a control methodology for wave energy converters which controls the velocity of the converter to ensure that it is in phase with the excitation force. In this section a model predictive controller is designed with the objective of tracking a reference trajectory which does the same. The magnitude of the trajectory to follow is defined as a constant scaling of the excitation force. This is a sub-optimal form of control.

4.1.1 Theory

Phase control is a common control methodology for wave energy converters found in literature (for example references [3], [8] and [10]). Ensuring that the velocity of the floater is in phase with the excitation force ensures that the excitation power (given as $P_e = \dot{z}F_e$) is always positive.

Specifying the phase only is not enough to define a suitable trajectory for the floater to follow. A suitable magnitude of the trajectory is more difficult to calculate. The following discussion briefly covers the reasoning behind selecting the resulting trajectory.

It is possible to express the magnitude of the floater trajectory which optimizes the mechanical power absorbed by the AWS. It should be stressed

that this expression does not take into account the power losses in the generator, assumes the motion is unconstrained, and assumes that system is linear. The velocity of the optimal trajectory is given as:

$$v_{opt}(\omega) = \frac{F_e(\omega)}{2R_i(\omega)} \tag{4.1}$$

where $R_i(\omega)$ is the intrinsic mechanical resistance. It is the sum of the resistance caused by the friction, the viscosity and the radiated wave. The theory behind equation 4.1 is given in chapter 4.3.2 of the literature study report and chapter 6 of reference [6].

Unfortunatly $R_i(\omega)$ is difficult to calculate. Additionally, to calculate the optimal velocity requires the Fourier transform of the excitation force. The Fourier transform of the excitation force can only be approximated because the future excitation force is not known. An approximate expression for equation 4.1 in the time domain is given as:

$$v_{opt}(t) \approx \frac{F_e(t)}{2d} \tag{4.2}$$

where d is the same average damping coefficient as used in the linear model.

Another disadvantage of the expression is that it assumes a power take off system with linear characteristics and does not take into account the generator losses. The generator losses are given as:

$$P_{gen} = c_1 \left(\frac{F_{gen}}{F_{rated}} \right)^2 + c_2 \left| \frac{\dot{z}}{v_{rated}} \right| + c_3 \tag{4.3}$$

where the first term represents the resistive losses and the second term represents the iron losses. Decreasing the magnitude of the velocity would decrease the iron losses. The resistive losses depend on the effort needed to follow the reference trajectory, which in turn depends on the difference between the natural response of the floater to the excitation force. By trial and error, a better reference trajectory for the given generator losses is given as:

$$v_{ref}(t) = \frac{F_e(t)}{2.5d} \tag{4.4}$$

Another problem with following a reference trajectory is that it can enter the constrained zone. To calculate the optimal trajectory which respects the constraints is more difficult. Model predictive control is capable of handling the constraints but it should be stressed that this does not mean that it handles the constraints optimally.

4.1.2 Controller design

The linear model used by the model predictive controller is derived in section 3.2. The general state space form is:

$$\dot{\vec{x}} = A\vec{x} + B_u F_{gen} + B_d F_e \tag{4.5}$$

The performance index is defined as:

$$\begin{aligned} J &= C_z\vec{x} + D_{zu}F_{gen} + D_{zr}v_{ref} \\ &= \begin{pmatrix} 0 & -1 \\ 0 & 0 \end{pmatrix}\vec{x} + \begin{pmatrix} 0 \\ \lambda \end{pmatrix}F_{gen} + \begin{pmatrix} 1 \\ 0 \end{pmatrix}v_{ref} \\ &= \begin{pmatrix} v_{ref} - \dot{z} \\ \lambda F_{gen} \end{pmatrix} \end{aligned} \tag{4.6}$$

where λ is a weight on the control force. The weight prevents the controller from tracking the reference over-zealously which could cause the closed loop to become unstable and would increase the generator losses. Since F_{gen} is in the order of 10^5 N and accepting a tracking error in the order of 10^{-1} m, the weight is set to $\lambda = 10^{-6}$.

Given the initial state and the future excitation force one can use the above equations to express the future performance index and the future state as an affine function of the future control force. The predicted performance index ($\hat{J}$) can then be optimized with respect to the control signal ($\hat{F}_{gen}$).

$$\underset{\hat{F}_{gen}}{min} \quad \hat{J}^T\hat{J} = \underset{\hat{F}_{gen}}{min} \quad \tfrac{1}{2}\hat{F}_{gen}^T H \hat{F}_{gen} + \hat{F}_{gen}^T f \qquad s.t.\ K\hat{F}_{gen} \leq l \tag{4.7}$$

where H and K are constant matrices, and f and l are constant vectors. $\hat{J}$ and $\hat{F}_{gen}$ are vectors containing predictions of the performance index and control force at discretized time steps until the horizon is reached. The horizon is defined as the length of time that the controller 'looks' into the future. The procedure of formulating the optimization problem is described in appendix D and reference [1].

The reference velocity is defined by equation 4.4, except that the maximum reference velocity is limited to the rated velocity.

The horizon of the controller is chosen as the natural period of the AWS. The step size is chosen so that there are 30 steps within the horizon. The constraints of the optimization problem are set to prevent the floater position from exceding the water break limit and the floater velocity from exceding the rated velocity. The m-code which implements the controller is given in appendix B.2, where the standard predictive control toolbox [1] is used.

4.1.3 Results

Figure 4.1 shows the performance for different synthetically generated sea states. The power spectra used to generate the sea states are made to resemble those at the test site. For each sea state the closed loop is simulated

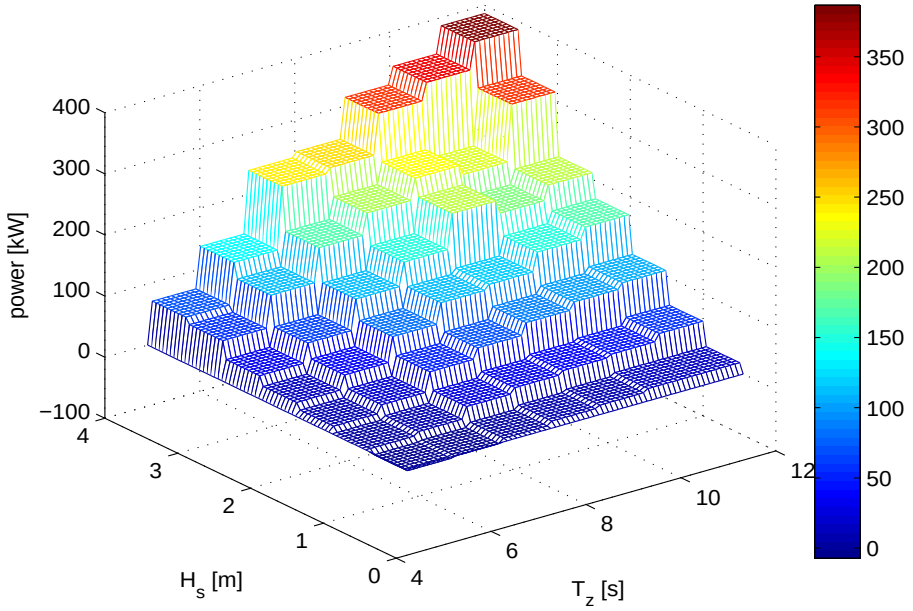

Figure 4.1: Average power produced for synthetic sea states over 7 minute intervals

for 7 minutes, and the average power produced is calculated. The power produced increases strongly with the significant wave height and also increases with the wave period.

The average power produced is irregular. This is because the waves are irregular and the simulation time is short. If the simulation time is increased the effects of the irregularities should disappear. Unfortunatly due to time constraints the simulation lengths were limited to 7 minutes.

A typical result is discussed. The significant wave height is 2.25 m and the average wave period is 9.5 s. Figure 4.2 shows the reference tracking performance of the closed loop. The floater follows the reference quite well. The control force remains moderate, except for where the optimization problem becomes infeasible.

Figure 4.3 shows the energy and power produced. The power produced is irregular. The power produced is often negative when the control force does not damp the floater velocity, but rather accelerates it.

Figure 4.4 shows how the controller is influenced when the wave force is not known a priori, for 9 different sea states. The figure shows the average power produced in each sea state for 3 types of controllers. Type 1 controllers are ideal controllers where the excitation force is known a pri-

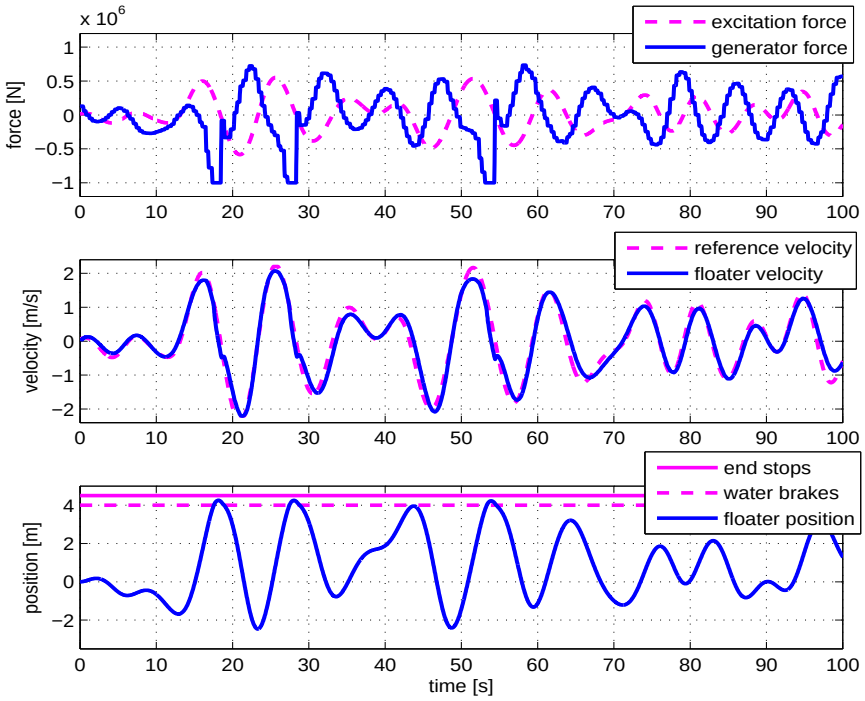

Figure 4.2: Tracking performance ($H_s = 2.25$ m and $T_z = 9.5$ s)

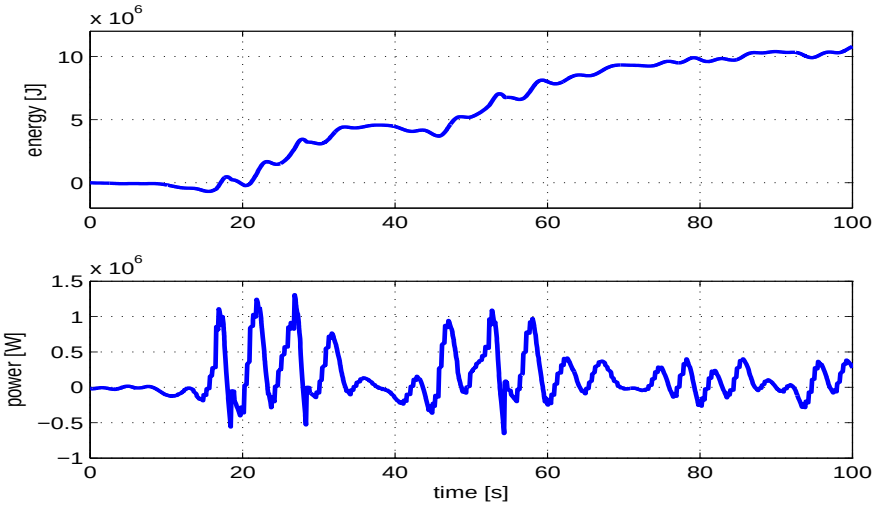

Figure 4.3: Energy performance ($H_s = 2.25$ m and $T_z = 9.5$ s)

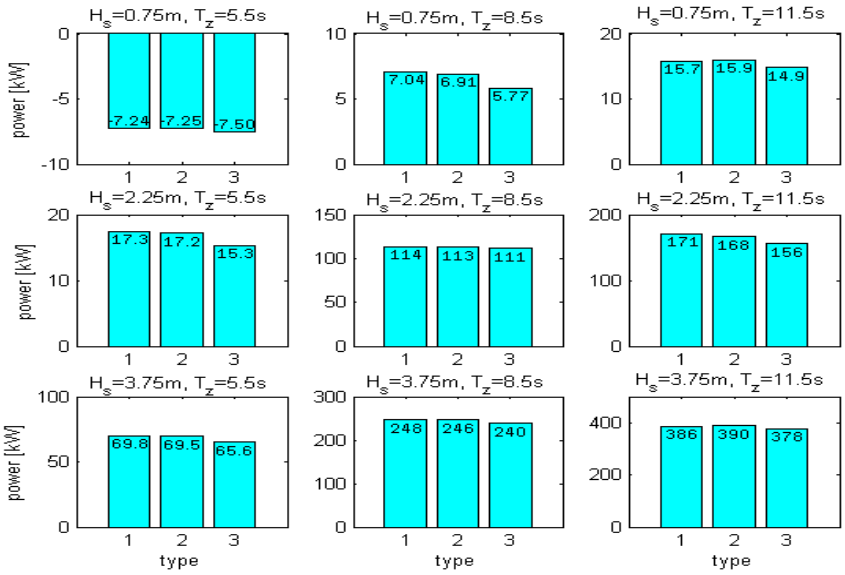

Figure 4.4: Comparison of average power produced when less information is available to the controller

ori. Type 2 controllers make use of an AR model to extrapolate known values of the excitation force to provide an estimate of the future excitation (see section 3.3). Type 3 controllers first estimate the excitation force from the motion of the device (see section 3.4), and then uses the obtained values to estimate the future excitation. In general type 2 controllers show a marginally worse performance than the type 1 controllers. In a few cases the type 2 controller performs marginally better. Type 3 controllers often show a significantly performance, typically by about 5 to 10 %.

4.2 Direct Energy Maximization

The energy produced by the floater can also be maximized in a more direct manner by calculating the control force based on an optimization procedure which attempts to maximize an expression for the energy produced. In this manner the generator losses can easily be taken into account, and the energy produced can be maximized with respect to the constraints.

4.2.1 Theory

The energy produced by the AWS can be approximated as a function of the floater velocity ($\dot{z}$) and the generator force (F_{gen}):

$$E = \int_{t_0}^{t_{final}} \left(-\dot{z}F_{gen} - c_1 \left(\frac{F_{gen}}{F_{rated}} \right)^2 - c_2 \left| \frac{\dot{z}}{v_{rated}} \right| - c_3 \right) dt \tag{4.8}$$

where the first term $\dot{z}F_{gen}$ represents the mechanical power absorbed by the generator, and the other terms represent the generator losses.

The velocity is a function of the control force as well as the excitation force and initial state. Given the initial state and a prediction of the excitation force it is possible to write an expression for the predicted energy with the control force as the only variable. It is then possible to formulate an optimization problem which finds the control force that maximizes the energy produced.

Unfortunatly the optimization problem is non-linear which is a difficult optimization problem to solve. It has high computational demands and may contain local maxima. Reference [5] provides an overview of non-linear optimization problems.

To investigate the optimization problem further the problem is solved for 12 different cases. For each case the optimization problem is formulated in terms of different future excitation forces and different initial conditions. Each of these optimization problems is then solved for 200 different initial guesses of the generator force signal. The generator force predictions are vectors where each element is a random value between $-F_{rated}$ and F_{rated}. For moderate and large excitaion forces the optimization algorithm always finds the same solution. This is encouraging because it shows that the optimization problem has no or very few local optima. For small excitation forces the optimization problem gives some variation in the solutions found. These seas states however, are not very interesting because there is little energy available anyway. Appendix E gives a more elaborate overview of the random search results.

4.2.2 Controller Design

The designed controller uses the matlab function *fmincon.m* to minize the equation:

$$\begin{array}{llll} min & (-E) = & min & \sum_{i=1}^{N} \left(\dot{z}[i+0.5]\hat{F}_{gen}[i] + c_1 \left(\frac{\hat{F}_{gen}[i]}{F_{rated}} \right)^2 + c_2 \left| \frac{\dot{z}[i]}{v_{rated}} \right| \right) \\ \hat{F}_{gen} & & \hat{F}_{gen} & \end{array} \tag{4.9}$$

where the velocity is predicted using a linear model (see appendix D). The velocity values denoted by $\dot{z}[i+0.5]$ in equation 4.9 are taken as the average

between two consecutive points (i.e. $\dot{z}[i+0.5] = 0.5(\dot{z}[i] + \dot{z}[i+1])$. This is to minimize the innacuracies caused by evaluating the cost function at discrete points only.

The optimization problem often becomes infeasible for high-energy sea states when hard constraints are applied to restrict the motion. For this reason soft constraints are applied for larger floater deviations, whereby a penalty is added to the cost function 4.9 when the prediction of the state approaches the water brake limit.

The penalty is added for each step whereby the absolute position exceeds 3.5 m or the absolute floater velocity exceeds the rated velocity. The magnitude of the penalty is given as:

$$Pen = \begin{cases} 0, & |z| < 0 \\ 4 \cdot 10^6 \cdot \left((|z| - 3.5)^2 + (|\dot{z}| - v_{rated})^2 \right), & \dot{z} \geq 0 \end{cases} \tag{4.10}$$

Additionally, hard constraints are implemented to prevent the floater from hitting the end stops.

The horizon is chosen as one natural period. A horizon of one natural period is long enough to ensure that the controller attempts to maximize the overall energy produced instead of attempting to maximize the short term power produced while neglecting the long term concerns. The step size should be small enough to minimize the effects of the model-mismatch. The stepsize is chosen so that there are 30 steps within the horizon. Appendix B.2 contains the m-code used to implement the controller.

4.2.3 Results

Figure 4.5 shows the performance for different synthetically generated sea states. The power spectra used to generate the sea states are made to resemble those at the test site. The power produced increases strongly with the significant wave height and also increases with the wave period.

The average power produced is irregular. This is because the waves are irregular. If the simulation time is increased the effects of the irregularities should disappear. Unfortunatly due to time constraints the simulation lengths were limited to 7 minutes.

A typical result is discussed. The significant wave height is 2.75 m and the average wave period is 8.5 s. Figure 4.6 shows the closed loop response. The reference velocity used in the reference tracking controller is also shown for comparison. The velocity has a similar magnitude to the reference, but is slightly out of phase.

The controller has no problem handling the constraints in this particular sea state. Note the steepness of the control force at 75 seconds in figure

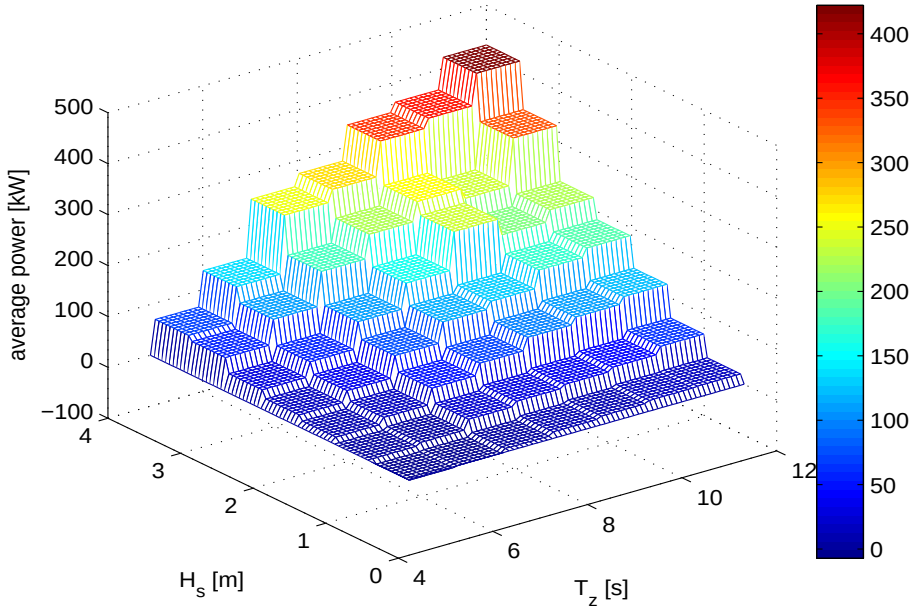

Figure 4.5: Average power produced for synthetic sea states over 7 minute intervals

4.6. This shows how the controller applies a sudden extra damping to handle the constraints. Section 5.3 provides a more general discussion on the constraint obedience.

It is observed that the generator force does not only apply a damping to the floater, but also accelerates it sometimes. Accelerating the floater *requires* energy, which the generator pulls from the electrical grid. The reason that this happens is because the controller does not attempt to maximize the instantaneous power produced but maximizes the energy produced over a certain time interval. The controller uses energy from the grid to excite the floater in order to maximize the energy produced on the long run. In fact, what the controller does is ensures that the closed loop is more in resonance with the excitation, despite the frequency of the excitation being irregular.

Figure 4.6 shows the energy and power produced. The power produced fluctuates and is often negative. Nevertheless the net energy produced steadily increases.

Figure 4.8 shows how the controller is influenced when the wave force is not known a priori, for 9 different sea states. The figure shows the average power produced in each sea state for 3 types of controllers. Type 1 controllers are ideal controllers where the excitation force is known a priori.

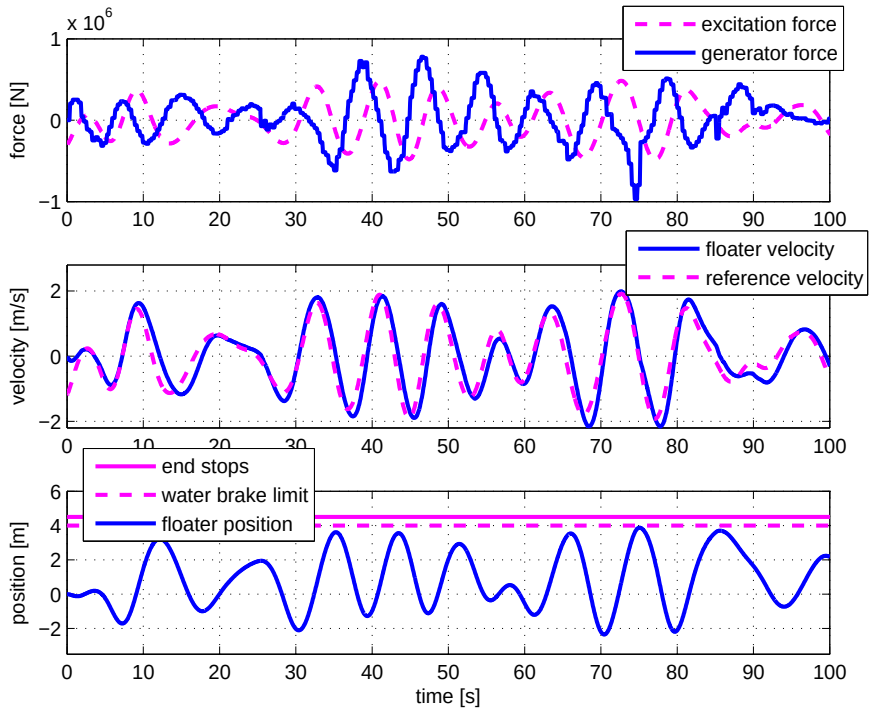

Figure 4.6: Closed loop response ($H_s = 2.75$ m and $T_z = 8.5$ s)

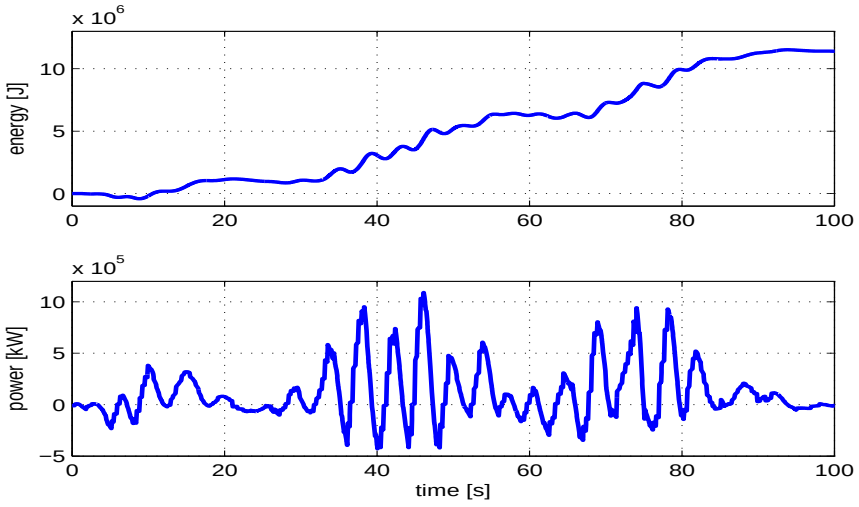

Figure 4.7: Energy performance ($H_s = 2.75$ m and $T_z = 8.5$ s)

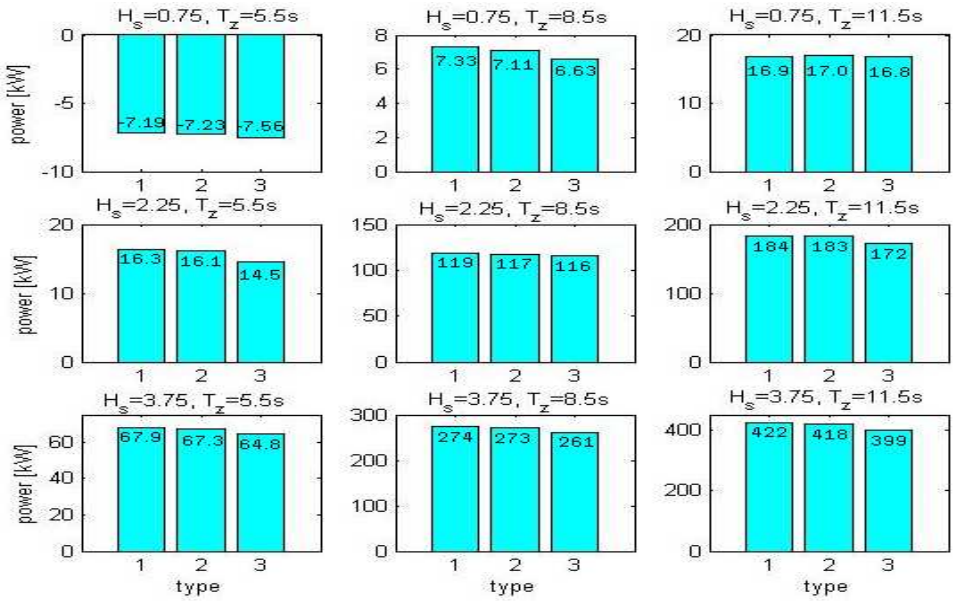

Figure 4.8: Energy produced for various sea states

Type 2 controllers make use of an AR model to extrapolate known values of the excitation force to provide an estimate of the future excitation (see section 3.3). Type 3 controllers first estimate the excitation force from the motion of the device (see section 3.4), and then uses the obtained values to estimate the future excitation. The type 2 controllers perform marginally worse than the type 1 controllers, typically by about 1 to 2 %. Type 3 controllers also perform marginally worse than the type 2 controllers, also typically by about 1 to 5%.

4.3 Switching between the Controllers

The reference tracking controller has problems in dealing with the constraints, and the energy maximizing controller is computationally expensive. Combining the two controllers could improve the overall performance.

4.3.1 Theory

The reference tracking controller is used to control the device under normal operating conditions. When the trajectory approaches the constraints the energy maximizing controller can be used with a smaller horizon. The idea is to let the energy maximizing controller handle the contraints only. This

has the advantage that a smaller horizon can be used, hence lightening the computational requirements.

4.3.2 Controller Design

The controller is designed such that the controller switches from the reference tracking problem to the energy maximization problem when the prediction of the floater position exceeds the water brake limit (κ). The controller switches back from the energy maximizing problem to the reference tracking problem when the any part of the floater position prediction is under a position 2 m below the water brake limit *and* at least two seconds have passed since the energy maximizing controller was switched on. In this manner the controller does not switch between the problems too frequently.

When the controller is solving the reference tracking problem and the difference between the reference velocity and actual velocity is large, a decaying exponential function is added to the reference velocity. This is done to ensure a smooth transition between the two controllers. The exponential function used is given as follows:

$$e = (\dot{z}(t_{switch}) - v_{ref}(t_{switch}))e^{t_{switch}-t} \tag{4.11}$$

where t_{switch} is the time at which the reference tracking controller was switched on.

The horizon of the reference tracking controller is set to one tuning period and the horizon of the energy optimizing controller is set to one third of a tuning period. The step size is selected to so that there are 30 steps within one tuning period.

Appendix B.2 contains the code used to implement the controller.

4.3.3 Results

Figure 4.9 shows the improvement of the average power produced of the switching controller over the reference tracking controller. In low-energy sea states the controller never switches to the energy maximizing controller and the average electrical power produced is the same. In high energy sea states an improvement is observed of at most 10 kW. The improvement is good, but does not approach the same performance as the energy maximizing controller (see figure 5.1).

An example for a high-energy sea state is discussed. The significant wave height is 3.75 m and the average wave period is 8.5 s. Figure 4.10 shows the closed loop response. When the floater approaches the constraints the controller switches to an energy maximizing tactic.

It has been demonstrated that the switching controller has potential to improve the energy production of the reference tracking controller while

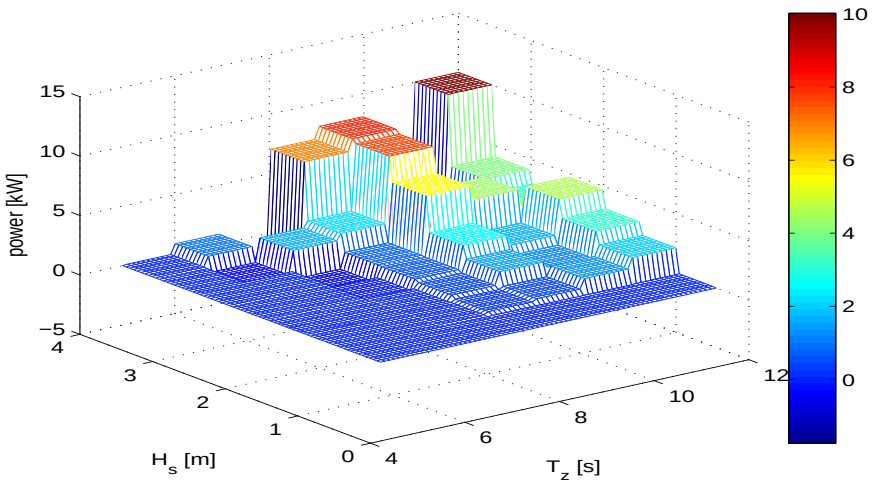

Figure 4.9: Improvement over the reference tracking controller

retaining relatively low computational requirements. If ultimately the computational requirements of the energy maximizing controller prove too high the switching controller provides a good alternative

The controller is parameterized by a number of extra tuning parameters:

- when to switch to the energy maximizing controller
- when to switch back to the reference tracking controller
- the energy maximizing horizon

Further research would be necessary to fine-tune the controller to maximize the increase in power given the processing availability.

4.4 Approximating the Energy Function for Faster Optimization

The expression representing the energy produced as a function of the floater velocity and generator force is non-linear. Additionally, the floater velocity depends on the generator force. This makes finding the optimal generator force a computationally expensive optimization problem. This section discusses one possibility of approximating the cost function to allow faster optimization.

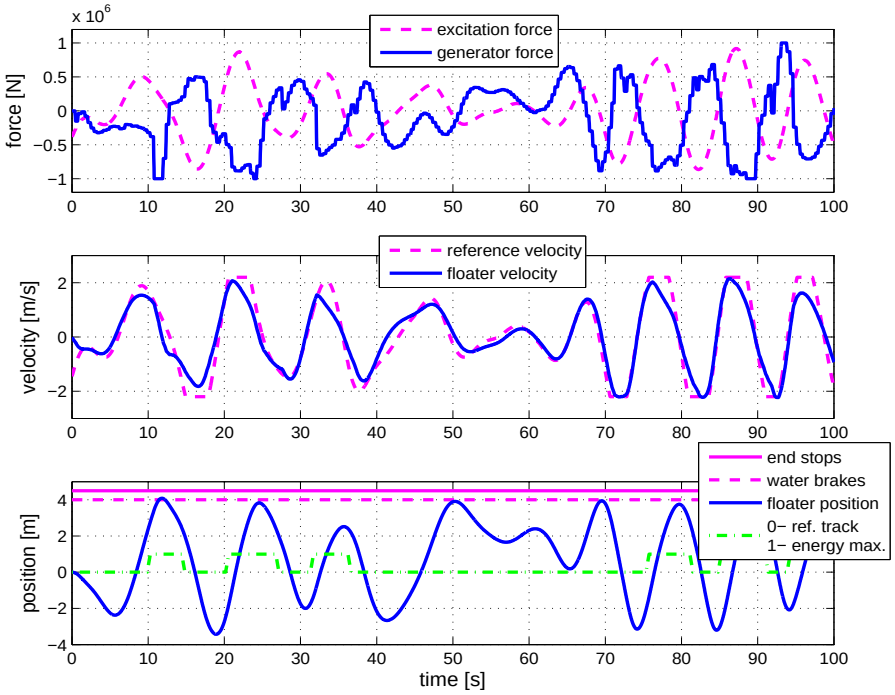

Figure 4.10: Closed loop response ($H_s = 3.75$ m and $T_z = 11.5$ s)

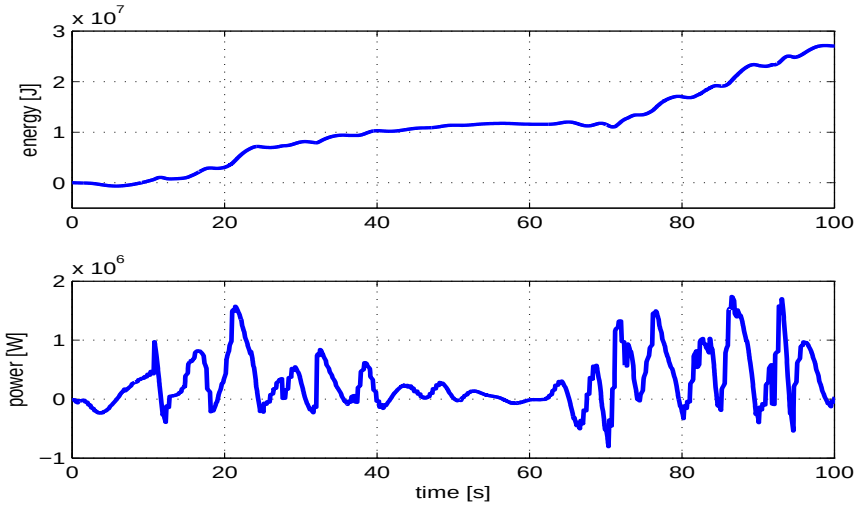

Figure 4.11: Energy performance ($H_s = 3.75$ m and $T_z = 11.5$ s)

4.4.1 Theory

The main problem with the controller that maximizes the energy is that the optimization problem is non-linear, making it computationally expensive. It is possible to construct an approximate expression for the energy in a more convenient form for optimization. For example, the cost function:

$$J = 2\lambda F_{gen}\dot{z} + (\lambda F_{gen})^2 + \dot{z}^2 \tag{4.12}$$

is similar in form to equation 2.4, and the optimization problem is a quadratic problem. Implementing this controller gives positive results but significantly worse than the reference tracking controller and is not mentioned further.

Another option to provide an approximate expression for the power produced is to first solve the reference tracking problem, and then to formulate another optimization problem making the predicted control force and the predicted velocity attained from the reference tracking problem. The procedure is described as follows:

Step 1: Given the predicted excitation force and the initial state the reference tracking optimization problem is solved. This yields a computed control force ($\hat{F}_{gen0}$) and an associated predicted velocity ($\hat{v}_0$).

Step 2: The change in velocity and change in control force are defined as:

$$\begin{aligned} \delta\hat{v} &= \hat{v} - \hat{v}_0 \\ \delta\hat{F}_{gen} &= \hat{F}_{gen} - \hat{F}_{gen0} \end{aligned} \tag{4.13}$$

where $\hat{F}_{gen}$ is the new control force. The new predicted velocity ($\hat{v}$) is a function of the new control force aswell as the initial state and the predicted excitation force.

Step 3: Substituting $\hat{v} = \delta\hat{v} + \hat{v}_0$ and $\hat{F}_{gen} = \delta\hat{F}_{gen} + \hat{F}_{gen0}$ into the power equation:

$$\begin{aligned} \hat{P} =& -(\delta\hat{v} + \hat{v}_0)(\delta\hat{F}_{gen} + \hat{F}_{gen0}) - c_1\left(\frac{\delta\hat{F}_{gen} + \hat{F}_{gen0}}{F_{rated}}\right)^2 - c_2\left|\frac{\delta\hat{v} + \hat{v}_0}{v_{rated}}\right| - c_3 \\ =& \ \delta\hat{v}\delta\hat{F}_{gen} - \hat{F}_{gen0}\delta\hat{v} - \hat{v}_0\delta\hat{F}_{gen} - \hat{v}_0\hat{F}_{gen0} \\ & - \frac{c_1}{F_{rated}^2}\left(\delta\hat{F}_{gen}^2 + 2\hat{F}_{gen0}\delta\hat{F}_{gen} + \hat{F}_{gen0}^2\right) - c_2\left|\frac{\delta\hat{v} - \hat{v}_0}{v_{rated}}\right| - c_3 \end{aligned} \tag{4.14}$$

The change in power is then given as (assuming that $\delta\hat{v} + \hat{v}_0$ has the same

sign as $\hat{v}_0$):

$$\delta\hat{P} = \quad \delta\hat{v}\delta\hat{F}_{gen} - \hat{F}_{gen0}\delta\hat{v} - \hat{v}_0\delta\hat{F}_{gen} - \frac{c_1}{F_{rated}^2}\left(\delta\hat{F}_{gen}^2 + 2\hat{F}_{gen0}\delta\hat{F}_{gen}\right) - c_2\frac{\delta\hat{v}}{v_{rated}} \tag{4.15}$$

Step 4: Assuming that the terms $\delta\hat{v}\delta\hat{F}_{gen}$ and $\delta\hat{F}_{gen}^2$ are small we get:

$$\delta\hat{P} \approx -\left(\hat{F}_{gen0} + \frac{c_2}{v_{rated}}\right)\delta\hat{v} - \left(\hat{v}_0 + c_1\frac{2\hat{F}_{gen0}}{F_{rated}^2}\right)\delta\hat{F}_{gen} \tag{4.16}$$

Both $\delta\hat{v}$ and $\delta\hat{F}_{gen}$ can be expressed as affine functions of $\hat{F}_{gen}$. Hence the *increase* in power over the reference tracking problem can be expressed as an affine function of the generator force.

To implement the controller first the reference tracking problem is solved, yielding the vectors $\vec{v}_0$ and $\vec{F}_0$. Next a linear system is formulated which describes $\begin{pmatrix}\hat{F}_e^T & \vec{v}_0^T & \vec{F}_{gen0}^T & \hat{F}_{gen}^T\end{pmatrix}^T \rightarrow \begin{pmatrix}\delta\hat{v}^T & \delta\hat{F}_{gen}^T\end{pmatrix}^T$. The matlab function *linprog* is used to determine the generator force which maximizes equation 4.16. The variables $\delta\hat{v}$ and $\delta\hat{F}_{gen}$ are constrained because equation 4.16 is only valid for small changes. Appendix B.2 contains the code used. Reference [5] provides an overview of linear programming problems.

4.4.2 Results

The optimization problem itself performs better. Unfortunately, when implemented in closed loop, the power produced is less than for the reference tracking controller. An improvement can only be seen when the reference to follow is chosen poorly. Figure 4.12 shows the closed loop response of the reference tracking controller when the reference is chosen as $v_{ref}(t) = F_e(t)/(4 \cdot d)$, and the improvement which results from the second optimization step. As a comparison the figure also shows the power produced when the reference is chosen well ($v_{ref}(t) = F_e(t)/(2.5 \cdot d)$).

When the reference is chosen well there is usually a slight degradation in the power produced. Even though the method does not yet provide an improvement to the overall results, it shows some promise in optimizing the energy produced while retaining low computational demands. The method is recomended for further research.

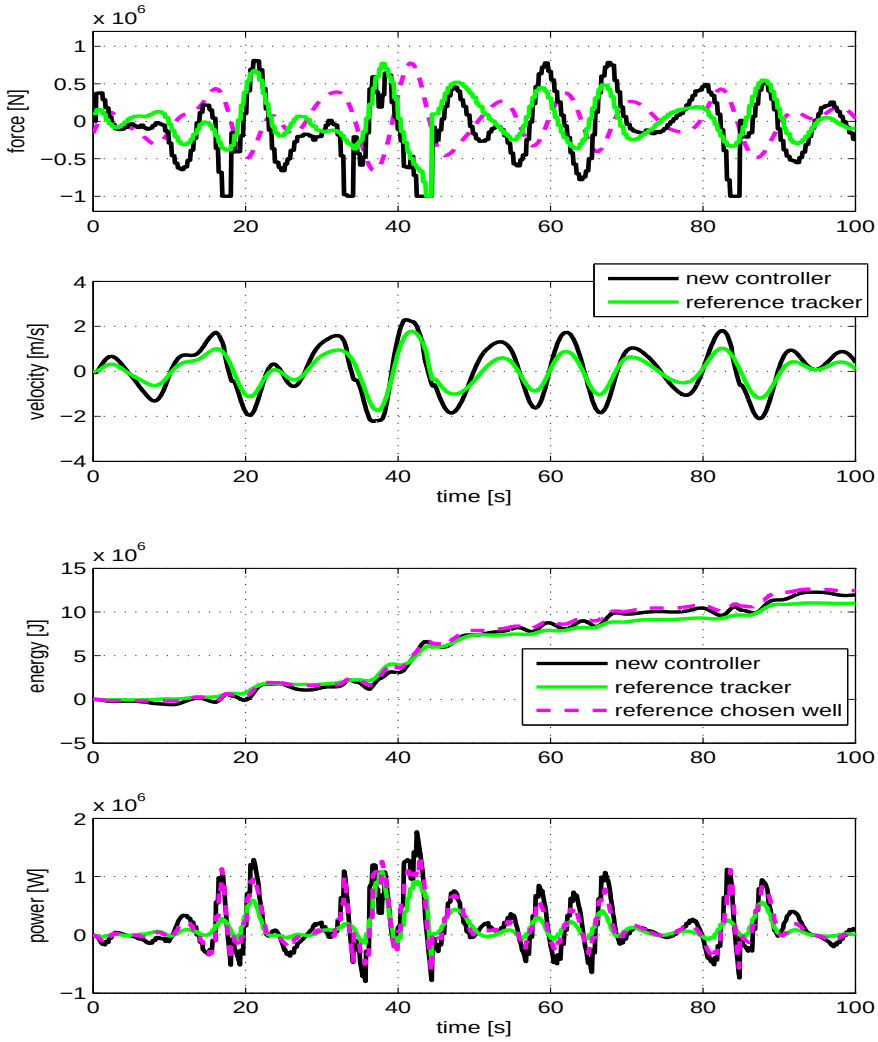

Figure 4.12: Improvement in closed loop performance ($H_s = 2.75$ m and $T_z = 9.5$ s)

Chapter 5

Evaluation

This chapter discusses the performance of the controllers. Section 5.1 compares the reference tracking controller and the energy maximizing controller. Section 5.2 discusses how close the controllers are expected to come to producing the maximum power available. Section 5.3 investigates the ability of the controller to handle the constraints. Section 5.4 investigates the effect of the uncertainties and approximations made.

5.1 Comparison of the Controllers

The energy maximizing controller and reference tracking controller produce similar amounts of energy. Figure 5.1 shows the difference in the energy produced between the two. As can be seen the energy maximizing controller has performs better in high energy sea states where the constraints play a major role.

Figure 5.2 and 5.3 show the closed loop response to a chirp excitation signal, with a magnitude of $0.5 \cdot 10^6$ N with a frequency which increases linearly from 0.25 rad/s to 1 rad/s. The tuning frequency of the AWS is set to 0.5 rad/s.

For lower frequencies the reference velocity guides the floater into the constraints. The reference tracker has trouble dealing with this and stutters around the water brake limit. The energy maximizing controller also approaches the constraints but handles them much more smoothly. The power produced in both cases is relatively low.

Around the tuning frequencies the power produced is relatively high. The reference tracking controller no longer has to deal with the constraints. The energy maximizer controller still pushes the floater close to the constraints and extra breaking is needed. Interesting to note is that the power output of the reference tracking is lower when the floater is at its peak than at its height.

For high frequencies the reference tracking controller needs more and

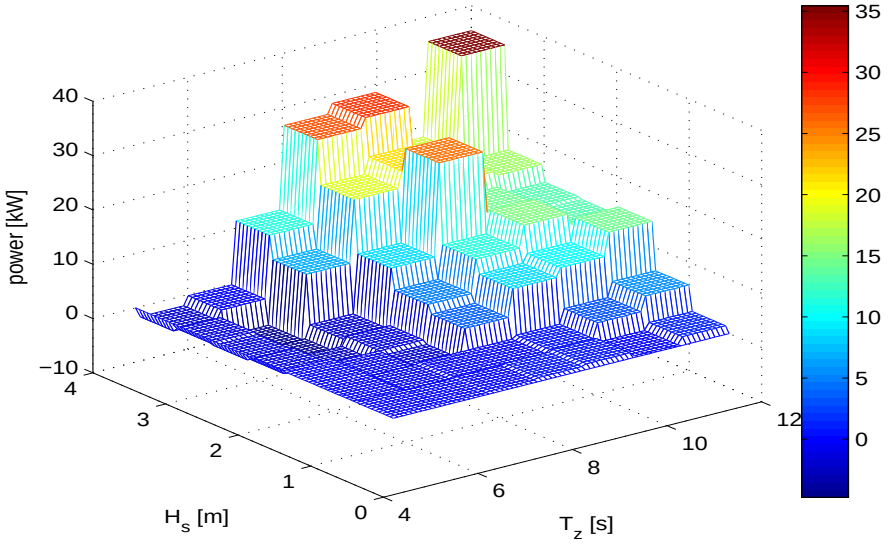

Figure 5.1: Improvement of the energy maximizer over the reference tracker

more effort to keep the floater on reference. The power produced in both cases decreases significantly.

Figures 5.4 up to 5.6 show the power breakdowns for three sea states. The graphs compare the average power entering the system from the waves (P_e), the average mechanical power absorbed by the generator (P_{mech}), and the average electrical power produced by the generator (P_{elec}). The graphs also show the mechanical losses.

In general the energy maximizing controller yields a higher excitation power, but has larger mechanical losses. For sea states with high energy the energy maximizing controller gains more by increasing the excitation than it loses by the higher mechanical losses, when compared to the reference tracking controller. For sea states with little energy the reference tracking controller and the energy maximizing controllers produce about the same average electrical power.

The reference tracking controller does not handle the constraints well. The relatively high average power lost by the water brakes reflects this.

5.2 Optimality

It is very difficult to estimate the maximum amount of energy that can be produced given a certain excitation. Work to make such an estimate

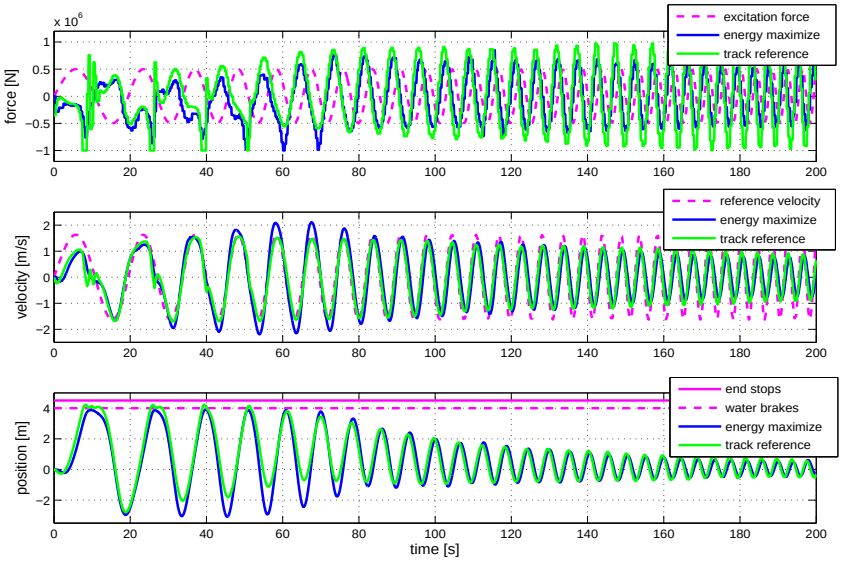

Figure 5.2: Response to a chirp excitation signal

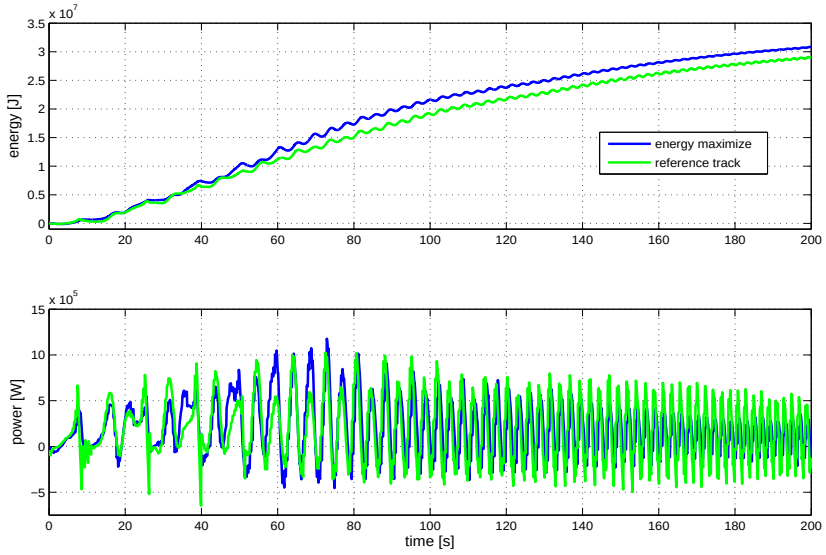

Figure 5.3: Electrical energy and power produced for a chirp excitation signal

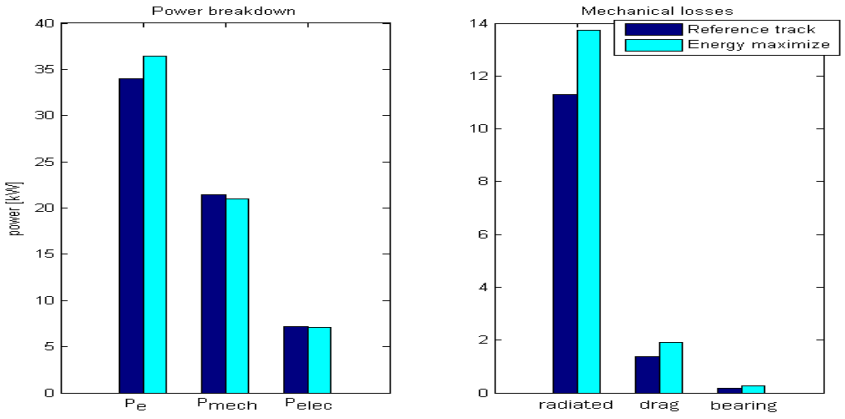

Figure 5.4: Power breakdown for $H_s = 1.25$m and$T_z = 6.5$s

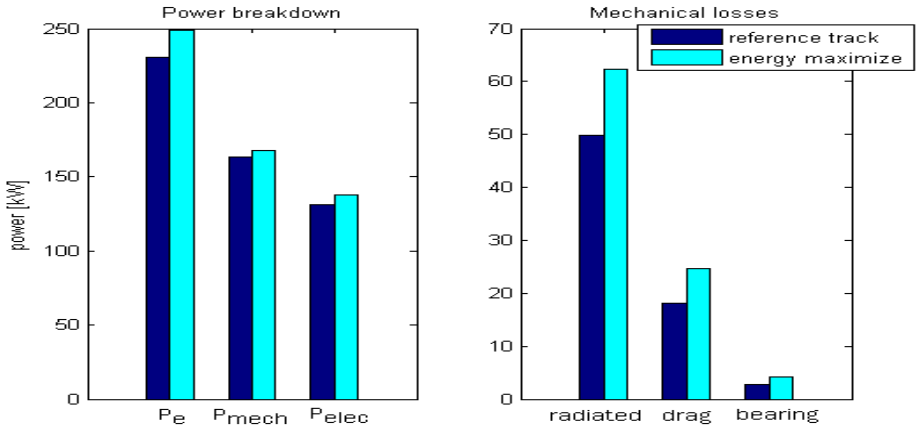

Figure 5.5: Power breakdown for $H_s = 2.25$m and$T_z = 8.5$s

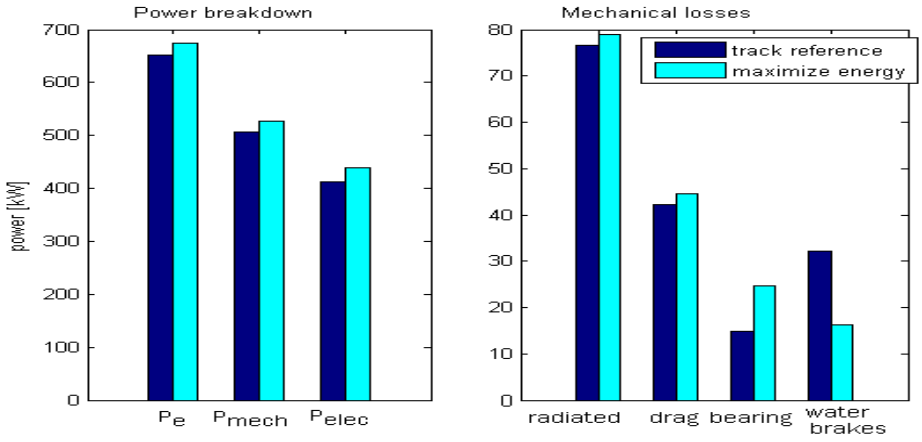

Figure 5.6: Power breakdown for $H_s = 3.75$m and$T_z = 11.5$s

has been done for systems which are linear and which have an unlimited stroke, as done for example by Falnes [6]. When the constraints play a larger role however, optimality becomes a very complicated issue.

It seems highly unlikely that the reference tracking controller performs close to the optimum because of the simplified way in which the reference trajectory is defined. By keeping the floater velocity in phase with the excitation force the controller deals with the irregular wave periods. The excitation power can be maximized by maximizing the velocity magnitude, but this would increase the losses. Figure 5.7 shows the power breakdown for three different reference velocities. When the magnitude is large the excitation power is large, but the mechanical and generator losses are also large.

The problem is that a constant scaling factor is assumed. For the true optimal trajectory the scaling factor actually varies with the frequency of the excitation and is influenced by the non-linearity of the device.

The controller which attempts to maximize the energy produced in principal should be close to optimal because the linear damping accounts well for the losses. The performance is limited by the following factors:

- model-mismatch between the linear model and the non-linear model causes an innacurate velocity
- the penalty on large floater positions and velocities interefere with the cost function
- the cost function is the sum of the power evaluated at discretized point instead of the integral of the power over time

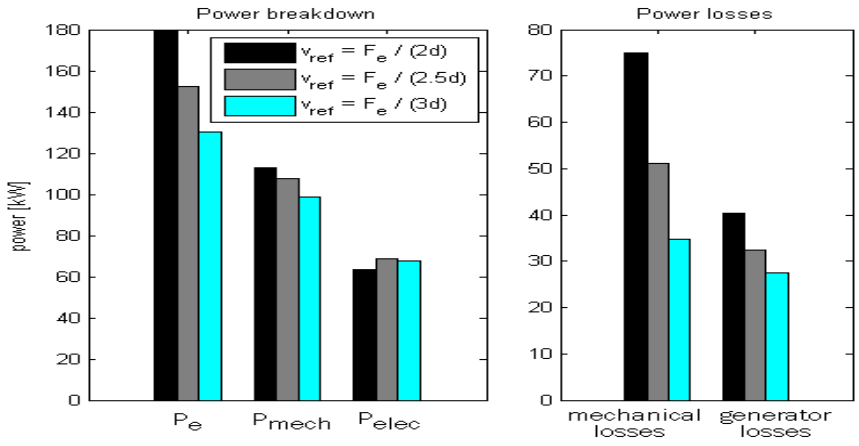

Figure 5.7: Effect of changing the magnitude of the reference velocity ($H_s = 1.75$m and$T_z = 8.5$s)

The fact that the performance is similar to the reference tracking controller suggests that the controller can still be improved.

Further research is needed to provide an estimate of the optimal performance.

5.3 Constraint Handling

One of the control challenges is to ensure that the constraints are met for all sea states for which the AWS is active. The closed loop simulation is run for the synthetically generated sea states and its constraint obedience is checked. The significant waves heights span from 0.75 m to 3.75 m and increase in steps of 0.5m. The average zero-crossing periods span 5.5 s to 11.5 s, and increase in steps of 1 s. Table 5.1 shows the maximum velocity at which the end stops are hit over simulation lengths of 7 minutes.

Energy maximize		
H_s [m]	T_z [s]	velocity [m/s]
3.75	9.5	0.5984
3.75	11.5	0.8906

Reference track		
H_s [m]	T_z [s]	velocity [m/s]
3.25	9.5	0.2647
3.75	9.5	0.7732
3.75	10.5	0.1555
3.75	11.5	0.8309

Table 5.1: Constraints breach

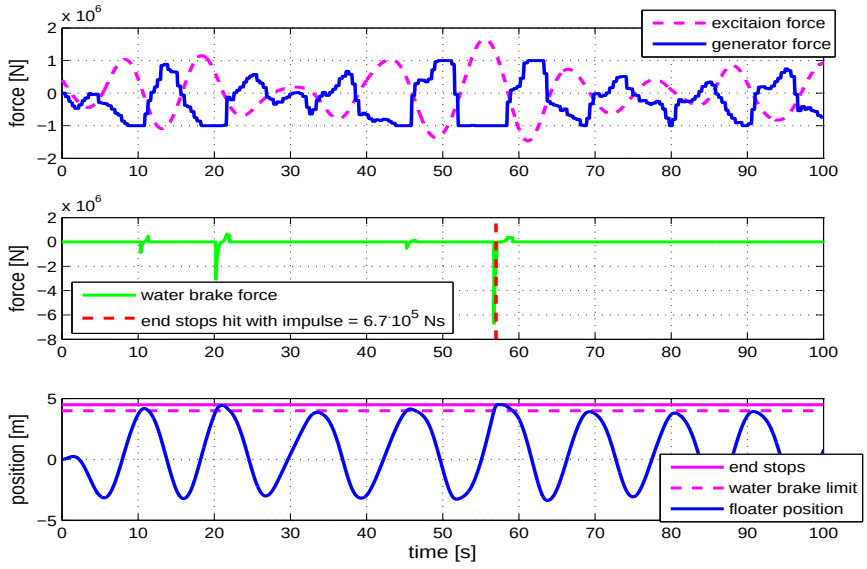

Figure 5.8: Closed loop response of the AWS for a high-energy sea state

In order to investigate further, figure 5.8 shows the closed loop response for a sea state with $H_s = 3.75$ m and $T_z = 11.5$ s. As can be seen the controller reacts to the incoming constraint by setting the controller force to maximum far in advance. Unfortunately the excitation force is simply too large for the generator to cope with, and the constraints are breached. It is possible to detune the AWS to make constraint handling easier. Figure 5.9 compares the closed loop responses in the cases that the AWS is well-tuned and detuned. In the detuned case the constraints are easily met, at the cost of the energy produced.

5.4 Effect of Assumptions

The results presented in this report is based on a number of assumptions. The following assumptions were made to simplify the control problem:

- the velocity and acceleration signals are known to the controller
- the non-linear stiffness of the floater is exactly known
- the generator losses are known
- the tide level is always equal to zero

Furthermore, in order to simplify simulating the system the hydrodynamic forces acting on the floater are calculated in advance given the wave profile for a floater at mid-position.

This section discusses the expected implications of the above mentioned assumptions to the performance. The assumptions made to construct the non-linear model are not discussed. Section 5.4.1 discusses the implications for the estimation of the excitation force. Section 5.4.2 discusses the implications for the closed loop performance.

5.4.1 Excitation Estimation

To estimate the excitation force it was assumed that the non-linear spring characteristics are known, and that the position, velocity and acceleration of the floater are known. The effect of using a simplified spring model, and poor signals for the velocity and acceleration of the floater on the excitation signal is investigated.

The stiffness is highly non-linear and varies with the floater position. The stiffness also varies with the tuning frequency. Assuming that the stiffness is known exactly would be quite optimistic. Instead the stiffness is approximated as an affine function of the floater position, parameterized by the tuning stiffness (k_n):

$$k(z) = 1.15k_n - 140\sqrt{k_n} \cdot z \tag{5.1}$$

Figure 5.10 shows the results for different tuning frequencies, whereby the dashed lines are given by equation 5.1 is used to estimate the spring force, and the solid lines show the stiffness used in the non-linear model. The approximation 5.1 is basically chosen quite arbitrarily in order to get a feel for the effects model-mistmatch on the excitation estimation. Figure 5.11 compares in the estimate of the excitation force when the exact stiffness is used and when the stiffness is approximated. There is little difference between the two results.

Assuming that the spring force as used in the non-linear model gives a good representation of the actual spring force, and that the real spring force can be approximated to a similar degree, it can be concluded that the excitation force can be reasonably estimated from the motion of the device, given reliable values for the velocity and acceleration.

The velocity of the floater is used to estimate the damping forces. The damping forces are already estimated in a rough way and are relatively small compared to the excitation force. A velocity signal can be obtained from the current and voltage information, or by filtering the position and/or the acceleration signals if available. The acceleration of the floater is needed to estimate the excitation force. The acceleration signal can be attained by filtering the position and/or velocity signals or incorporating an inertial sensor.

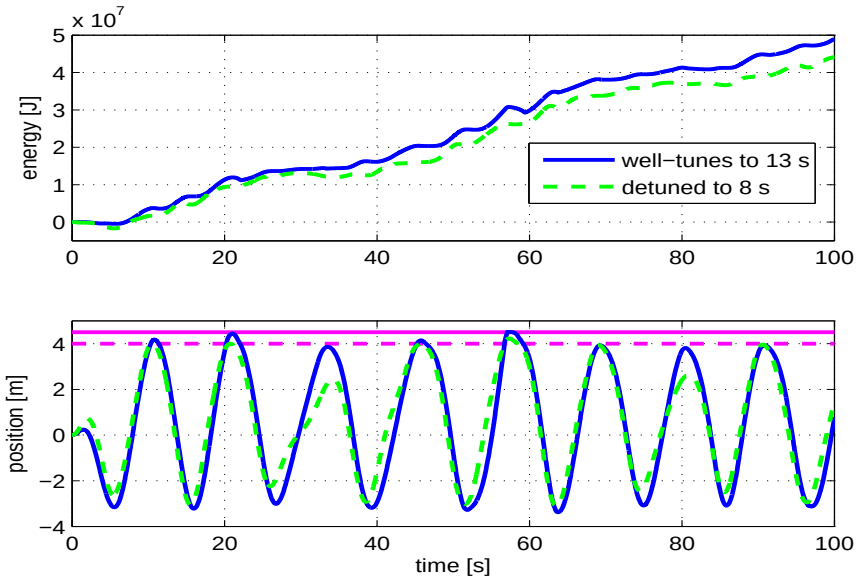

Figure 5.9: Comparison of the well-tuned and de-tuned responses

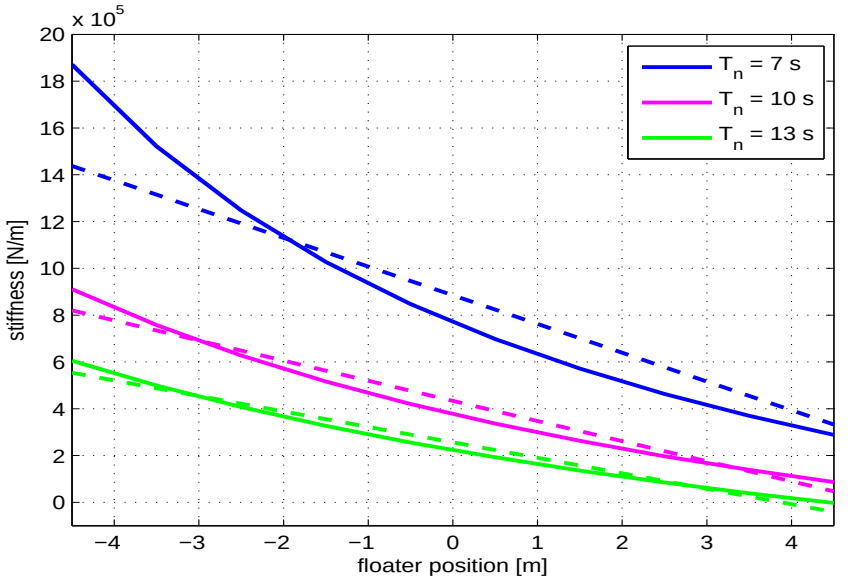

Figure 5.10: Stiffness approximated as an affine function of floater position

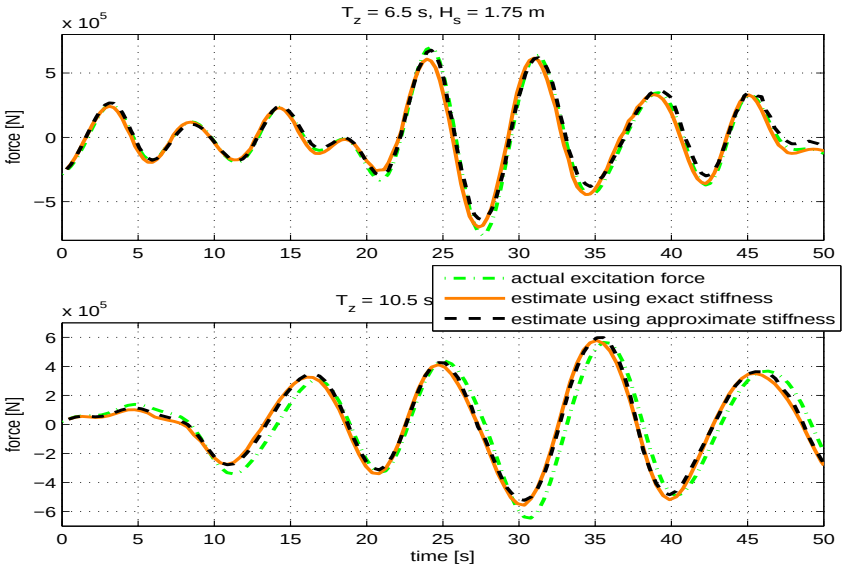

Figure 5.11: Excitation estimate using exact and approximate stiffness models

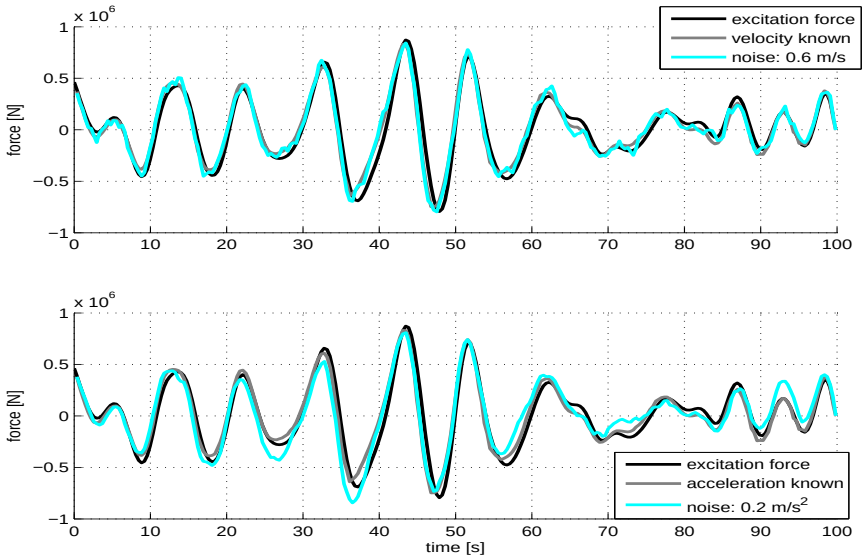

Figure 5.12: Effect of poor velocity and acceleration signals on excitation estimate ($H_s = 2.25$ m, $T_z = 8.5$ s)

In order to evaluate how the quality of the signals affects the performance a noisy signal is superimposed on the acceleration signal used by the estimator. The noisy signal is parameterized by it's magnitude and has a bandwidth of 2π rad/s. Figure 5.12 shows the results for varying magnitudes of noise added to the velocity and acceleration signals. The noise added to the velocity signal has marginal impact on the estimate. The noise added to the acceleration has a much larger impact.

When the excitation is estimated poorly, the closed loop would be influenced. Figure 5.13 shows the influence of adding noise to the acceleration, and using equation 5.1 to approximate the stiffnes on the closed loop performance. It can be concluded that a noise component on the acceleration signal of be larger than 0.2 $\mathrm{m/s^2}$ significantly decreases the closed loop performance.

This investigation does not conclusively determine the minimum requirements on the quality of the signal for input estimation.

5.4.2 Closed Loop Performance

The following simplyfying assumptions are relevant to the closed loop performance, given an estimate of the excitation force:

- The calculation of the excitation force is approximated by calculating it for a floater at mid-position.
- The tide level was assumed zero.
- The velocity signal was assumed known.
- The generator losses are assumed known.

Excitation force approximation: The vertical hydrodynamic force acting on the floater is calculated from the surface elevation profile for the floater at mid-position. This signal is used as the excitation force. This is an approximation because the hydrodynamic force depends on the depth and hence on the floater position. It is lower when the floater position is low, and higher when the floater position is high.

The vertical hydrodynamic pressure is given as:

$$p_e(d,\omega) = \rho g \eta(\omega) K_P(d,h,\omega) \tag{5.2}$$

where $\eta(\omega)$ is the surface elevation in the frequency domain and $K_P(d,h,\omega)$ is the decay factor which depends on the depth to the sea bed (h) and the depth at which the pressure is felt (d). Figure 5.14 shows how the decay factor varies with depth and frequency. This means that for high floater positions the hydrodynamic force is larger, and for low floater positions the hydrodynamic force is lower.

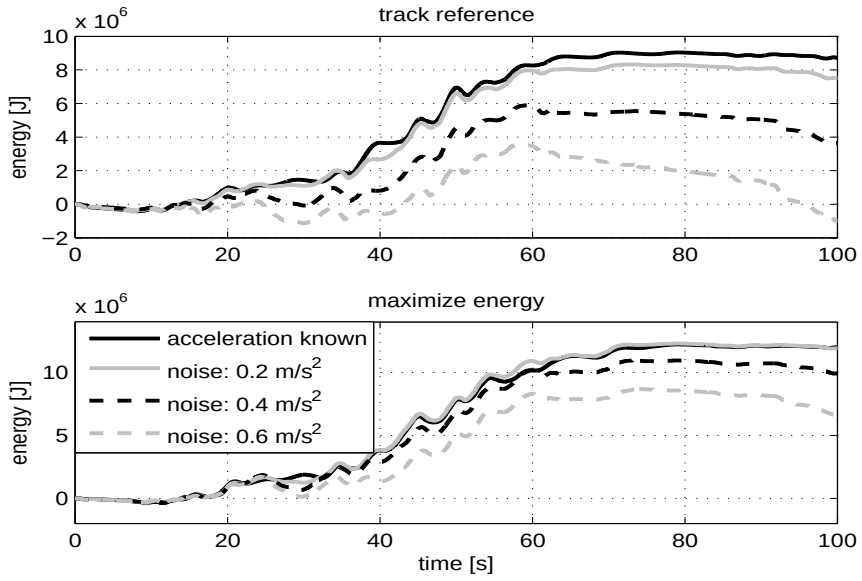

Figure 5.13: Approximating stiffness and adding noise to the acceleration signal ($H_s = 2.25$ m, $T_z = 8.5$ s)

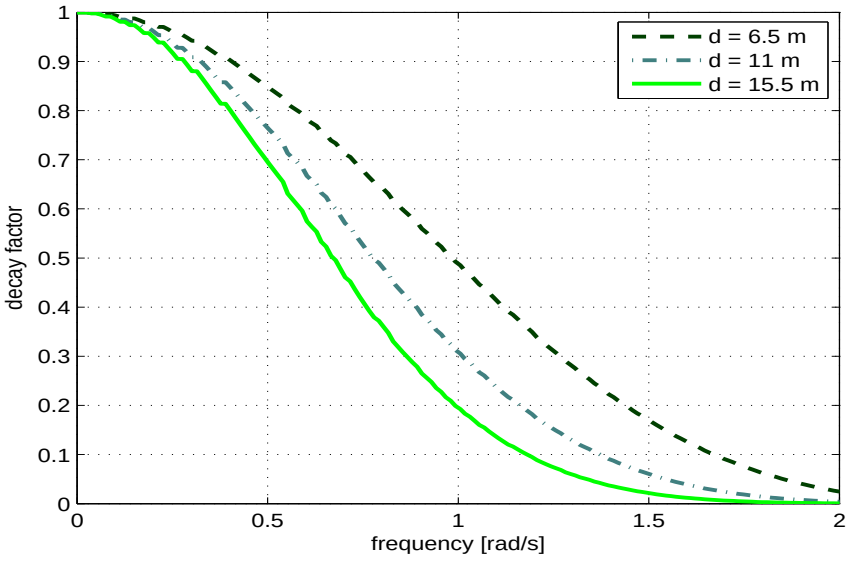

Figure 5.14: Decay factor for different depths

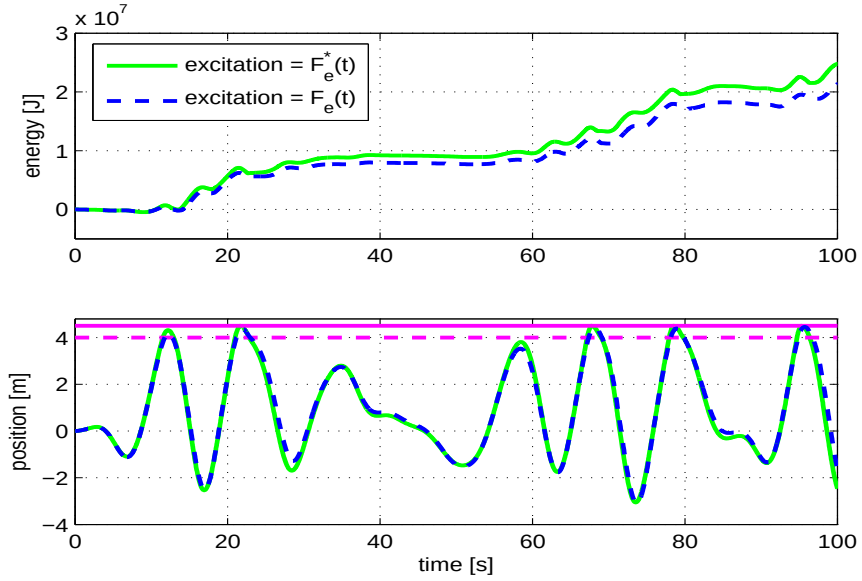

Figure 5.15: Influence of the changing floater depth ($H_s = 2.75$ m, $T_z = 10.5$ s)

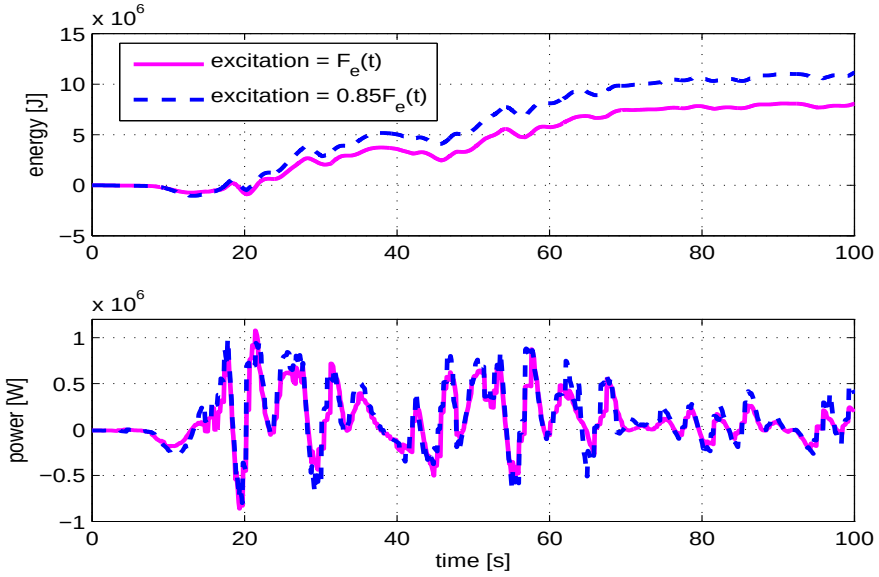

Figure 5.16: Influence of the decay factor on the energy produced ($H_s = 2.25$ m, $T_s = 9.5s$)

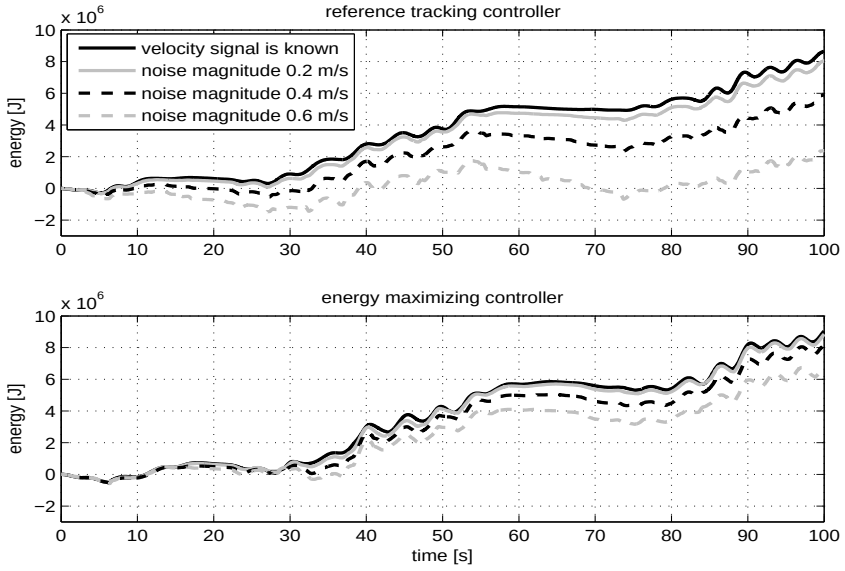

Figure 5.17: Performance for poor quality velocity signals ($H_s = 2.75$ m, $T_z = 8.5$ s)

To investigate the effect of this approximation the excitation force is varied as a function of the floater position as:

$$F_e^* = F_e(0.25z + 1) \tag{5.3}$$

which means that $F_e^* = 1.5F_e$ when the floater is at its maximum height and $F_e^* = 0.5F_e$ when the floater is at its lowest height. This is quite a high approximation of how the excitation force is likely to change.

Figure 5.15 shows the influence on the performance. In this case the energy produced increases, but the constraints become more difficult to obey. The energy produced increases because the hydrodynamic force is larger for high floater positions where the stiffness is smaller and it is smaller for low floater positions where the stiffness is larger.

Tide level: The tide level was assumed equal to zero. For higher tide levels the depth increases which decreases the vertical hydrodynamic force and the energy produced is expected to decrease. The factor by which the excitation force decreases depends on the frequency. To get a rough idea of the expected decrease in performance we compare two cases

controlled by the energy optimizing controller using the excitation force calculated for $H_s = 2.25$ m, $T_s = 9.5$ s and $\eta_T = 0$. For the two cases the excitation force is multiplied by a factor 1 and 0.85. The factor 0.85 is representative of an increase in tide level to 3 m, for a frequency of 0.65 rad/s.

Figure 5.16 shows the results. The total electrical energy produced over the 100 second time period decreases from about 11 MJ to about 8 MJ. **Velocity signal:** The velocity signal is assumed to be available. In the final design of the AWS the quality of the velocity signal may not be known. In order to evaluate the effect of only having a poor quality velocity signal noise was added to the velocity signal used by the controller. Figure 5.17 shows how the noise affects the closed loop response. A noise component of larger than 0.2 m/s has significant impact on the performance. The reference tracking controller seems to be more sensitive to poor velocity signals than the energy maximizing controller.

Generator losses: The energy maximizing controller uses the same equation to maximize the energy produced as the non-linear model uses. When the generator losses are not known exactly there may be some degradation in performance. For example, if the generator losses are over estimated the generator would provide less effort to facilitate the motion of the device and the mechanically absorbed power would decrease. If the resistive losses are under estimated the mechanical power absorbed would increase, but the generator losses would also increase.

To give a rough idea of the expected losses the results are compared using an exact expression for the generator losses to the results when the generator losses are mis-estimated, for a sea state with $H_s = 2.25$ m and $T_z = 8.5$ s. Doubling the resistive loss coefficient used by the controller causes the reduces the average electrical power produced from 59 kW to 52.6 kW. When the controller under estimates the resistive losses by 75% the average electrical power produced decrease from 59 kW to 58.6 kW. When the controller ignores the generator losses and only attempts to maximize the mechanical power absorbed the average electrical power produced decreases to 42 kW.

Chapter 6

Conclusions

The control objective is to design a controller for the AWS which maximizes the power produced while obeying certain pre-set constraints. The control problem is made challenging by the ocean waves being highly irregular. It was opted to make use of a model predictive control strategy because of the importance of the constraints. Two controllers were designed. One controls the trajectory of the floater to ensure that the motion remains in-phase with the excitation force. The other attempts to maximize the power produced. The following conclusions can be drawn from the thesis study:

- The excitation force can be predicted by making use of a linear auto regressive model fitted to past data. The model is used to extrapolate future values from the most recent. The closed loops are simulated for both the cases whereby the future excitation force is known a-priori, and whereby the excitation force is predicted by the described method. Comparing the results show that predicting the excitation force causes only a small reduction in the performance. It is sufficient to predict the excitation force in this manner for this type of model predictive controller.

- A linear model is used by the model predictive controller, while the system is non-linear. Additionally, the damping is chosen in an approximate way, and is expected to show variation depending on the sea state. The linear model is accurate enough for the purpose of controlling the trajectory of the floater. Whether the linear model is sufficient to maximize the energy produced has not been confirmed.

- In order to estimate the excitation force from the motion of the floater by the method used in this report the quality of the acceleration signal is important. It has been demonstrated that when noise with a magnitude of larger than 0.2 m/s^2 is added to the acceleration signal a significant reduction in the closed loop performance is likely.

- The constraints can be handled for most sea states, without any problem. In the most stormy sea conditions evaluaed the constraints are breached. In such cases it is possible to detune the AWS to allow it to stay operational.

Chapter 7

Reccomendations

The work done on control of the AWS by this thesis is far from exhaustive. There is still a lot of research that can be done. The following recommendations are made for future research.

- Controlling the trajectory of the floater so that it remains in phase with the excitation force is a control strategy frequently found in literature. However, the optimal gain is more complicated to calculate, especially when the motion is constrained. Further research could be done in calculating the optimal trajectory from the excitation force. Alternatively the optimal trajectory could be found by an adaptive algorithm.

- The controller which attempts to maximize the non-linear optimization problem performs well but has high computational requirements. The optimization problem should be further examined to be able to decrease the computation time needed. Alternatively one could do further research into approximating the energy produced to an expression which is easier to optimize.

- It is difficult to estimate the maximum amount of energy that can be produced, and whether the controllers come close to performing so well. The energy optimizing controller is expected to come close because the linear damping used to estimate the energy produced accounts well for the losses. Nevertheless further research is required.

- The report assumes the use of passive water brakes which activate automatically when the floater exceeds a certain position. It is a possibility to use active water brakes where the water brake action can be actively controlled. A control strategy whereby the water brake settings can be controlled should be designed and the improvement of the performance evaluated.

- The controllers designed require an excitation force signal. When the signal is inacurate, the prediction of the signal is also inacurate and the performance decreases. The signal could be provided by a pressure sensor which would decrease the reliability, or could be estimated from the motion of the floater which requires a good model of the AWS and a reliable signal of the floater acceleration. A trade-off could be made between these two options. Alternatively other options could be considered, such as using information from surface measurements or information from nearby AWSs to estimate the signal.

- The AWS is a non-linear system. A linear model is used to predict the motion of the floater which is in turn used to calculate the control force. If a more accurate model is used to predict the motion this may improve the performance, especially for high-energy sea states where water brakes are often applied. Using a more accurate model would raise the computational requirements.

- The report has focused on control of a single AWS. It is intended for the AWS to be deployed in wave parks numbering about 20. Distributed control and information exchange to improve the overall performance can be considered.

- The report has explored solely the possibility of using model predictive control. Other control strategies could also be investigated.

Appendix A

List of Symbols

c_1, c_2, c_3	power loss coefficients
C_{DDW}	drag coefficient when the floater velocity is downwards
C_{DUP}	drag coefficient when the floater velocity is upwards
d	linear damping coefficient
d_f	water depth
g	acceleration due to gravity
h_f	floater height
H_s	significant wave height
J	performance index
k	linear spring coefficient
$\bar{L}_G$	equivalent spring length
m	apparent mass of the floater
m_{add}	added mass: extra inertia felt by the floater due to the wave it radiates
m_f	floater mass
M_x	horizontal hydrodynamic moment
p_{amb}	ambient pressure
P	power produced by the AWS
P_e	excitation power
P_{gen}	generator power losses
P_{mech}	mechanical power losses
R_i	intrinsic mechanical resistance
S_f	inner cross-sectional area of the floater
S_F	outer cross-sectional area of the floater
t	time
T_z	average upwards-zero-crossing wave period
v_{opt}	optimal velocity
v_{rated}	designed maximum velocity
w	white noise
$\vec{x}$	state
z	vertical displacement of the floater

Table A.1: List of symbols: miscellaneous

A	state matrix
B_d	disturbance input matrix
B_u	control input matrix
C_y	output matrix to measured output
C_z	output matrix to performance index
D_{yd}	feedthrough matrix from disturbance to output
D_{yu}	feedthrough matrix from control signal to output
D_{zr}	feedthrough matrix from reference to performance index
D_{zu}	feedthrough matrix from control signal to performance index

Table A.2: List of symbols: state space matrices

F_{bear}	coulomb friction in bearings
F_d	damping force
F_{drag}	drag force
F_e	excitation force
F_{gen}	control force supplied by the generator
F_{grav}	force due to gravity
F_{hs}	hydrostatic force
F_k	spring force
F_{Ntop}, F_{Nbot}	horizontal force felt by the top and bottom bearings
F_{rad}	radiation force
F_{rated}	designed maximum generator force
F_{sp}	gas spring force
F_{wb}	force due to the water brakes
F_x	horizontal hydrodynamic force
M_x	horizontal hydrodynamic moment

Table A.3: List of symbols: forces

β_{wb}	water brake coefficient
γ	heat capacity ratio
ϵ	filtered white noise made to resemble the damping discrepancy
κ	water brake limit
μ	bearing friction coefficient
η	sea surface elevation
η_T	tide level
ρ	water density
ω	angular frequency

Table A.4: List of symbols: greek letters

$\hat{}$	prediction
$\vec{}$	vector
$\bar{}$	equilibrium value
$\dot{}$	time derivative

Table A.5: List of symbols: accents

Appendix B

Matlab Code

The closed loop is programmed into m-code. The code can be categorized into two types: the framework and the specific controllers. The framework, given in section B.1, calculates the response of the AWS to the given hydrodynamic forces and the control force calculated by the specific controllers and also provides a prediction of the excitation force. The various controllers, given in section B.2, calculate the control force as a function of the current state and the prediction of the excitation force.

Additionally the code used to calculate the hydrodynamic loads acting on the AWS from the sea surface profile is given in section B.3.

B.1 Framework

The framework consists of 4 files:

- run_ model.m: the central m-file which runs the other files and simulates the closed loop using an ode solver.
- wave_ model.m: identifies an AR model used to predict the excitation force.
- Feedback.m: the ode solver calls the function after every successful integration step. It predicts the excitation force and calls the controller function.
- AWS.m: describes the dynamics of the AWS and calculates the estimated excitation force.

A number of MPC controllers have been implemented. Each uses two files identified by the suffix. The suffix can be defined on line 58 of *run_model.m*. The files are *formulate***.m* and *controller***.m*. The first defines the necessary parameters, the second is called by *Feeback.m* and calculates the control input.

B.1.1 Central m-file

```
%
% The central m-file simulates the closed loop system by the following
% steps:
%
%   (1) initialize: loads the wave data, defines AWS parameters, formulates
%       the optimization problem for control etc...
%   (2) wave prediction model: identifies a model used to predict the wave
%       excitation force
%   (3) run simulation: simulates the closed loop by running two functions
%           (a) d(state)/dt=AWS(t,state) describes the dynamics of the AWS
%           (b) Feedback(t,state) provides a prediction of the excitation
%           force and uses it to calculate a control signal
%
% * Select dataset: lines 29 and 30 (datasets contain excitation force and
%       horizontal loads; to use other values see lines 70 to 80)
% * Change controller parameters: lines 45 to 49
% * Specify controller: line 58
% * Change simulation parameters: lines 62 and 67

clear all;
close all;
clc;
addpath c:\paul\matlab71\toolbox\mpc2;

%(1) initialize
disp('initializing...');

%   -load wave data
load increasing_heights;
n=5;
global Tn;
Tn=T(n)*1.3;    %tuning period
mess=[];
if Tn>13
    message=sprintf(' * Tn too large. Tn set equal to 13 seconds\n'); disp(message);
    Tn=13; mess=1;
elseif Tn<7
    message=sprintf(' * Tn too small. Tn set equal to 7 seconds\n'); disp(message);
    Tn=7; mess=1;
end;

%   -AWS parameters
define_parameters;

%   -controller parameters
global N Nm lambda dt cont_type
dt=Tn/30;               %step size
N=round(Tn/(dt));       %prediction horizon
Nm=1;                   %minimum control horizon
%the type of controller used specified by the suffix of the m-files
%example: for reference tracking controller cont_type=''
%         for constrained reference tracking controller cont_type='_c'
```

```
%   no suffix       -reference tracking controller
%   _c              -constrained reference tracking controller
%   _e              -reference tracking controller including noise model
%   _2              -cost function = (velocity + lambda*Fgenerator)^2
%   _f              -full cost function = c1*(Fgen/Frated)^2 + c2*(abs(vel/v_rated))
cont_type=sprintf('_fsc_obs');

%   -simulation parameters
global type tfinal;
tfinal=50;             %simulation length [s]
% select type to determine the information available to the controller
%   type=1: excitation force known a priori
%   type=2: excitation force measured at present
%   type=3: excitation force estimated through AWS motion
type=1;

%   -external parameters
global Fen Fxn Mxn damp damp2
Fen=Fe(:,n+1);          %vertical hydrodynamic force
Fxn=Fx(:,n+1);          %horizontal hydrodynamic force
Mxn=Mx(:,n+1);          %horizontal hydrodynamic moment
damp=5200*H(n)+87000;   %estimated average damping
damp2=damp*1.75;        %damping used to define reference velocity
%   -re-interp the forces
    tn=Fe(:,1);
    t_new=0:dt:tn(length(tn));
    Fen=spline(tn,Fen,t_new);
    Fxn=spline(tn,Fxn,t_new);
    Mxn=spline(tn,Mxn,t_new);

%   -feedback parameters
global Fgen Fe_hat t_prev x_est Fgen0;
Fgen=0;         %initial generator force (control input)
Fe_hat=[];      %prediction of excitation force
t_prev=0;       %output time after last successful iteration
x_est=zeros(5,1);

%   -simulation outputs
global Fe_est t2
t_error=[];     %times at which optimization problem is not solved
flag=[];        %problem with the optimization procedure (flag=-2: infeasible)
Fe_est=[];      %estimated excitation force
water=[];       %water brakes active? (0=false, 1=true)
P=[];           %power output
t_out=[];       %points in time at which power is output
t2=[];          %time output corresponding to step-size
t3=[];          %time output including intra-iteration
Fgenrem=[];     %generator force
v_est=[];       %estimated velocity
t4=[];          %time output for using estimator
p_est=[];       %estimated position
e=[];           %x-p_est
swit=[];        %switcher controller: on/off value
Pl_c1=[];       %copper losses
```

```
Pl_c2=[];       %iron losses
Pe=[];          %excitation power
Pl_drag=[];     %drag losses
Pl_rad=[];      %radiated power
Pl_bear=[];     %bearing losses
Pl_wb=[];       %water brake losses
Fgen_out=[];    %generator force
v_est=[];       %estimated velocity when using observer
message_out=sprintf('\nMessages:\n'); %output messages
if mess==1
    message_out=sprintf('%s%s',message_out,message);
end

%   -formulate optimization problem
dt_obs=0.1;         %observer step size
form=sprintf('formulate%s',cont_type);
eval(form);

%(2) prediction model based on observations
disp('creating wave prediction model...');

%   -prediction matrix
global Cp order;
order=40;
%quick=0: for type 3 run the AWS model to generated estimated excitation
%         force. then use the estimated to id the wave model
%quick=1: identifies the wave model based on known past excitation force
%         with some noise super-imposed
quick=1;

%   -parameters for simulink model on which the wave model is based (if
%    quick=0)
Wave_Force=[Fe(:,1) Fe(:,n+1)];
Mhor=[Mx(:,1) Mx(:,n+1)];
Fhor=[Fx(:,1) Fx(:,n+1)];
global Rss;
load convolution;
Rss=ss(Rss.a,Rss.b,[25 0],0);

[Cp,mod]=wave_model(Fen,t_new,N,H,dt,order,type,quick,Wave_Force,Mhor,Fhor);

%(3) simulate closed loop
disp('running simulation...');

options=odeset('OutputFcn',@Feedback,'AbsTol',1e-3,'MaxStep',dt_obs);
if dt<=dt_obs
    options=odeset('OutputFcn',@Feedback,'AbsTol',1e-3,'MaxStep',dt);
end
y0=[0; 0; 0; 0; 0; 0; 0; 0; 0; 0; 0; 0; 0];

[t,y]=ode45(@(t,y) AWS(t,y,Fxn,Mxn),[0 tfinal],y0,options);

%(4) results
if length(message_out)>11
```

```
    disp(message_out);
end;

figure(1);
subplot(211);
plot(t,y(:,9),'m-');
hold on;grid on;

subplot(212);
plot(t_out,P,'m-');
hold on;grid on;
legend('energy','power');
```

B.1.2 Parameters

```
%Parameter Definitions
global m_f m_add mu_bear C_M C_D dtop0 dbot0 rho_sea d_out h_f S_F C_DUP C_DDW;
global beta_wb_min beta_wb_max delta d_pp g x_0 d_ZH eta_T h d_0 p_amb S_f;
global gamma_air F_hs_equi F_sp_equi L_sp_equi khs ktot wb_limit end_stops;
global c1 c2 c3 Frated vrated;

m_f=4e5;            %floater weight [kg]

m_add=3.55e5;       %added mass [kg]

mu_bear=0.1;        %bearing friction coefficient
C_M=2;              %coefficients use to calculate horizontal wave forces
C_D=1;
dtop0=19;           %initial distance from floater bottom to top bearing
dbot0=10;           %initial distance from floater bottom to bottom bearing

rho_sea=1.025e3;    %sea water density [kg/m^3]
d_out=11;           %floater diameter [m]
h_f=28.5;           %floater height [m]
S_F=95;             %floater outer area [m^2]
C_DUP=0.2;          %upwards drag coefficient
C_DDW=0.4;          %downwards drag coefficient

beta_wb_min=1e3;    %water brake minimum damping coefficient [kg/m]
beta_wb_max=1.5e6;  %water brake maximum damping coefficient [kg/m]

delta=9;            %stroke

g=9.8;              %acceleration due to gravitiy [m/s^2]

x_0=17.75;           %distance mid-floater to sea bed at equilibrium
d_ZH=43;            %water depth wrt hydrographic zero
eta_T=0;            %tide level
h=d_ZH+eta_T;       %depth to sea bed
d_0=d_ZH-x_0+eta_T; %depth to mid-position
p_amb=1e5;          %ambient pressure [N/m^2]

S_f=79;             %floater inner area [m^2]
```

```
gamma_air=1.4;      %isothermal/adiabatic
Fg=-m_f*g;

%compute stiffness at equilibrium
F_hs_equi=-S_F*(rho_sea*g*(d_0-h_f/2)+p_amb)+(S_F-S_f)*(rho_sea*g*(d_0+h_f/2)+p_amb);
F_PTO_offset=0;
F_sp_equi=-Fg-F_hs_equi-F_PTO_offset;
L_sp_equi=gamma_air*F_sp_equi/((m_f+m_add)*(2*pi/Tn)^2+rho_sea*g*S_f);
khs=-rho_sea*g*S_f;
ksp=gamma_air*F_sp_equi/L_sp_equi;
ktot=ksp+khs;

wb_limit=4;
end_stops=4.5;

c1=25e4;
c2=2e4;
c3=1e4;
Frated=1e6;
vrated=2.2;
```

B.1.3 Wave model

```
function [Cp,mod]=wave_model(Fe,t,N,H,dt,order,type,quick,Wave_Force,Mhor,Fhor);
%
% [Cp,mod]=wave_model(Fe,t,N,H,dt,order,type,quick,Wave_Force,Mhor,Fhor)
%
% creates a prediction of the excitation force, by:
%       (1) identifying an ar model
%       (2) creating a prediction matrix
%
% inputs:
%   Fe,t                    -> force inputs to AWS for id at times t
%   N                       -> prediction horizon
%   H                       -> sig. wave height
%   dt                      -> controller step size
%   order                   -> ar model order
%   quick                   -> toggle simulink model on/off
%   Wave_Force, Mhor, Fhor  -> in case quick==0 these are needed to tun the
%                              simulink model
%
% outputs:
%   Cp  -> matrix such that (    y(k)   )       (     0      )
%                           (   y(k+1)  )       ( y(k-order) )
%                           (    ...    ) = Cp *(    ...     )
%                           (    ...    )       (   y(k-1)   )
%                           (   y(k+N)  )       (    y(k)    )
%
%   mod -> ar model in iddpoly format

global Rss;
%(1) identify the model
%   -resample
```

```
if type==3
    if quick==0
        clear F time;
        [t,ex,y]=sim('observer');
        F=y(:,4);
        time=0:dt:t(length(t));
        F=spline(t,F,time);
        F=F+rand(1,length(F)).*max(F)/10;
    else
        % Noise should be added to represent the noise in the excitation force
        % signal. It is needed to 'teach' the ar model that there is a high
        % -frequency component.
        time=0:dt:t(length(t));
        F=spline(t,Fe,time);
        F=F+rand(1,length(F)).*max(F)/10;
    end;
else
    time=0:dt:t(length(t));
    F=spline(t,Fe,time);
end;

%   -identify model
past=500;              %number of id values

dat=iddata(F(1:past)');
mod=ar(dat,order);
a=mod.a;

%(2) construct prediction matrix
B=[-a(2:order+1); [eye(order-1) zeros(order-1,1)]];
C=[1 zeros(1,order-1)];
Cptemp=[1 zeros(1,order-1)];
for i=1:(N-1)
    Cptemp=[Cptemp; C*B^i];
end;

%   -re-order the matrix
[l,w]=size(Cptemp);
Cp=[];
for i=0:w-1
    Cp=[Cp Cptemp(:,w-i)];
end;
```

B.1.4 Feedback

```
function status=Feedback(t,state,flag);
%
% (1) Calculates the predicted excitation force.
%       -type 1: uses the same pre-calculated values of the force as used
%         as input in AWS.m
%       -type 2: makes a prediction of the future values using only past
%         and present values of the same pre-calulated values as used as
%         input in AWS.m
```

```
%       -type 3: makes a prediction of the future values using past and
%         present values of the estimation of the excitation force
% (2) Calls the controller function to calculate the next control input
% (3) Calculates and records the power output
% (4) Displays progress

global type Fe_hat t2 Fgen tfinal Fe_est dt t_prev Fen Cp cont_type;
global c1 c2 c3 Frated vrated;
global m_f m_add mu_bear C_M C_D dtop0 dbot0 rho_sea d_out h_f S_F C_DUP C_DDW;
global beta_wb_min beta_wb_max delta d_pp g x_0 d_ZH eta_T h d_0 p_amb S_f;
global gamma_air F_hs_equi F_sp_equi L_sp_equi khs ktot wb_limit end_stops;
global N order Fen Fxn Mxn Tn

if length(t)>2
    % make a prediction of the excitation force
    %   type 1: force is known a priori
    %   type 2: prediction is based on exact measurements of past and
    %           present excitation force
    %   type 3: prediction is based on an estimation of the excitation
    %           force based on the AWS motion
    if type==1
    % type 1: the excitation force prediction equals the future excitation
    % force

    % Output future values
    [a,b]=size(Cp);
    a=N;
    b=order;
    Fe_hat=Fen((floor(t(length(t))/dt)+1):(floor(t(length(t))/dt)+a))';
    t2=[t2 t(length(t))];

    elseif type==2
    % type 2: the excitation force prediction is extrapolated from the past
    % and present excitation force

    % select past values excitation fore, and make a prediction
    [a,b]=size(Cp);
    if (floor(t(length(t))/dt))>=b
        Fe_past=Fen(floor(t(length(t))/dt)+2-b:floor(t(length(t))/dt)+1)';
        Fe_hat=Cp*Fe_past;
        t2=[t2 t(length(t))];
    else
        Fe_hat=Fen((floor(t(length(t))/dt)+1):(floor(t(length(t))/dt)+a))';
        t2=[t2 t(length(t))];
    end

    elseif type==3
    % type 3: the prediction of the excitation force is extrapolated from
    % an estimation based on the motion of the AWS.

    % when step number increases make a new estimate of the excitation
    % force: the output of the filter as described in the report, computed in AWS.m
    if floor(t_prev/dt)<floor(t(1)/dt)
        t2=[t2 t(1)];
```

```
        Fe_est=[Fe_est -0.4*state(7,1)-0.16*state(8,1)+ ...
          (-62.5*state(3,1)-78.12*state(4,1)+400*((m_f+m_add)*state(1,1)-state(6,1)))];
    elseif floor(t(1)/dt)<floor(t(2)/dt)
        t2=[t2 t(2)];
        Fe_est=[Fe_est -0.4*state(7,2)-0.16*state(8,2)+ ...
          (-62.5*state(3,2)-78.12*state(4,2)+400*((m_f+m_add)*state(1,2)-state(6,2)))];
    elseif floor(t(2)/dt)<floor(t(3)/dt)
        t2=[t2 t(3)];
        Fe_est=[Fe_est -0.4*state(7,3)-0.16*state(8,3)+ ...
          (-62.5*state(3,3)-78.12*state(4,3)+400*((m_f+m_add)*state(1,3)-state(6,3)))];
    elseif floor(t(3)/dt)<floor(t(4)/dt)
        t2=[t2 t(4)];
        Fe_est=[Fe_est -0.4*state(7,4)-0.16*state(8,4)+ ...
          (-62.5*state(3,4)-78.12*state(4,4)+400*((m_f+m_add)*state(1,4)-state(6,4)))];
    end;

    % Select past values of excitation and make prediction
    [a,b]=size(Cp);
    if length(Fe_est)>=b
        Fe_past=[Fe_est(length(Fe_est)-b+1:length(Fe_est))];
        Fe_hat=Cp*Fe_past';
    else
        Fe_hat=Fen((floor(t(length(t))/dt)+1):(floor(t(length(t))/dt)+a))';
    end
    end
    t_prev=t(length(t));

    %calculate control input
    cont=sprintf('controller%s(t,state,Fe_hat);',cont_type);
    eval(cont);

    % calculate the power out
    P=-Fgen*state(2)-c1*(Fgen/Frated)^2-c2*abs(state(2)/vrated)-c3;
    % output power and points in time
    assignin('base','varInBase',P);
    evalin('base','P(end+1)=varInBase;');
    assignin('base','varInBase',t(length(t)));
    evalin('base','t_out(end+1)=varInBase;');

    % output power losses
    n=round(t(1)/dt)+1;
    Fe=Fen(n);
    Fx=Fxn(n);
    Mx=Mxn(n);
    Pl_c1=c1*(Fgen/Frated)^2;
    Pl_c2=c2*abs(state(2)/vrated);
%   -Fwb: water brakes (external)
    if state(1)>wb_limit
        Fwb=-beta_wb_max*(state(2))^2*sign(state(2));
        wb=true;
    else
        Fwb=0;
        wb=false;
    end;
```

```
%   -Frad: force due to radiated wave
    omn=2*pi/Tn;
    if omn<0.75
        R=omn*6.4286e4-8.2143e3;
    elseif omn<0.89
        R=40000;
    elseif omn<1.08
        R=-5.2632e4*omn+8.6842e4;
    else
        R=-7.1429e4*omn+1.0714e5;
    end
    Frad=-2*R*state(2);
%   - bearing losses
    xtop=dtop0-state(1);
    xbot=dbot0;
    FNtop=(Mx-Fx)/(xtop-xbot);
    FNbot=(Fx-FNtop);
    Fbear=-mu_bear*sign(state(2))*(abs(FNtop)+abs(FNbot));
%   -Fdrag: viscous drag
    if state(2)>0
        Fdrag=-0.5*C_DUP*rho_sea*S_F*(state(2)^2);
    elseif state(2)<0
        Fdrag=0.5*C_DDW*rho_sea*S_F*(state(2)^2);
    else
        Fdrag=0;
    end;

    % Output detailed power info
    Pe=Fe*state(2);
    Pl_drag=-Fdrag*state(2);
    Pl_rad=-Frad*state(2);
    Pl_bear=-Fbear*state(2);
    Pl_wb=-Fwb*state(2);

    assignin('base','varInBase',Fgen);
    evalin('base','Fgen_out(end+1)=varInBase;');
    assignin('base','varInBase',Pe);
    evalin('base','Pe(end+1)=varInBase;');
    assignin('base','varInBase',Pl_drag);
    evalin('base','Pl_drag(end+1)=varInBase;');
    assignin('base','varInBase',Pl_rad);
    evalin('base','Pl_rad(end+1)=varInBase;');
    assignin('base','varInBase',Pl_bear);
    evalin('base','Pl_bear(end+1)=varInBase;');
    assignin('base','varInBase',Pl_wb);
    evalin('base','Pl_wb(end+1)=varInBase;');
    assignin('base','varInBase',Pl_c1);
    evalin('base','Pl_c1(end+1)=varInBase;');
    assignin('base','varInBase',Pl_c2);
    evalin('base','Pl_c2(end+1)=varInBase;');

    % check the constraints
    if state(1,1)>(end_stops-0.01)
        text=sprintf(' * !!end stops hit!! with a velocity of %0.5g m/s\n',state(2,1));
```

```
        disp(text);
        assignin('base','varInBase',text);
        evalin('base','message_out=sprintf(''%s%s'',message_out,varInBase);');
    end

    % display the progress
    if length(t)>2
    if floor(t(length(t)))>floor(t(length(t)-1))
        show=sprintf('initializing...\ncreating wave prediction model...
          \nrunnning simulation...\nprogress: %d%%',floor(t(length(t))/tfinal*100));
        clc;
        disp(show);
    end;
    end;
end;
status=0;
```

B.1.5 AWS dynamics

```
function [dstatedt]=AWS(t,state,Fxn,Mxn);
%
% Describes the dynamics of the AWS as a system of differential equations.
%
%   d(state)
%   --------    =   f(state) + Fexternal
%      dt
%
% where state=[x v D1 D2 I1 I2 HP1 HP2 E R1 R2 ... RN]'.
%
% The system has two states to describing the motion (position (x) and
% velocity (v)), 4 states (D1, D2, I1 annd I2) implement the filter (see
% report) to estimate the excitation force, 2 states act as a high-pass
% filter, one state represents the energy output (E), and the last N states
% can be used to represent the radiation force.
%
% syntax: dstate_dt=AWS(t,state)
% the external forces are defined globally under the names:
%   Fen, Mxn, Fxn   ->  vertical and horizontal hydrodynamic loads
%                       interpolated in 0.1 second intervals
%   Fgen            ->  control force supplied by generator

%AWS parameters
global m_f m_add mu_bear C_M C_D dtop0 dbot0 rho_sea d_out h_f S_F C_DUP C_DDW;
global beta_wb_min beta_wb_max delta d_pp g x_0 d_ZH eta_T h d_0 p_amb S_f;
global gamma_air F_hs_equi F_sp_equi L_sp_equi ktot wb_limit khs end_stops;
global c1 c2 c3 Frated vrated Tn;

%external variables
global Fen Fgen Fe_hat type dt damp;

%output variables
global water;
```

```
n=round(t/dt)+1;
x=state(1);
xdot=state(2);
Fe=Fen(n);
Mx=Mxn(n);
Fx=Fxn(n);

%output time
assignin('base','varInBase',t);
evalin('base','t3(end+1)=varInBase;');

%Forces
%   -Fgen: control force (external)
assignin('base','varInBase',Fgen);
evalin('base','Fgenrem(end+1)=varInBase;');

%   -Fwb: water brakes (external)
%       Water brakes are active if the floater position is larger than wb_limit as
%       defined in 'define_parameters.m'
if state(1,length(t))>wb_limit
    Fwb=-beta_wb_max*(state(2,length(t)))^2*sign(state(2,length(t)));
    wb=true;
else
    Fwb=0;
    wb=false;
end;
assignin('base','varInBase',wb);
evalin('base','water(end+1)=varInBase;');

%   -Frad: force due to radiated wave
%Frad=93.75*state(10); %second order
%Frad=33.58*state(12); %fourth order
omn=2*pi/Tn;
if omn<0.75
    R=omn*6.4286e4-8.2143e3;
elseif omn<0.89
    R=40000;
elseif omn<1.08
    R=-5.2632e4*omn+8.6842e4;
else
    R=-7.1429e4*omn+1.0714e5;
end
Frad=-2*R*state(2);
%Frad=-15000*2*2*pi/Tn*state(2);
%Frad=-2500*state(2);

%   -Fbear: coulomb friction in bearings    xtop=dtop0-state(1);
xbot=dbot0;
FNtop=(Mx-Fx)/(xtop-xbot);
FNbot=(Fx-FNtop);
Fbear=-mu_bear*sign(state(2))*(abs(FNtop)+abs(FNbot));

%   -Fdrag: viscous drag
if xdot>0
```

```
    Fdrag=-0.5*C_DUP*rho_sea*S_F*(xdot^2);
elseif xdot<0
    Fdrag=0.5*C_DDW*rho_sea*S_F*(xdot^2);
else
    Fdrag=0;
end;

%   -Fspring: spring force due to air-pressure and nitrogen cylinders
Fspring=F_sp_equi*((x/L_sp_equi)+1)^(-gamma_air);

%   -Fgrav: gravity
Fgrav=-m_f*g;

%   -Fhs: hydrostatic force
Fhs=-S_F*(rho_sea*g*(d_0-h_f/2-x)+p_amb)+(S_F-S_f)*(rho_sea*g*(d_0+h_f/2-x)+p_amb);

% estimate known forces
k_nonlin=((1+state(1,length(t))/L_sp_equi)^(0.5*(-gamma_air-1)))* ...
    (F_sp_equi*gamma_air/L_sp_equi)+khs;
F_known=Fgen+Fwb-damp*xdot-k_nonlin*x;

% end stop reaction force: should halt the floater with dt_es seconds
if x>end_stops && xdot>0
    dt_es=0.01;
    Fes=-xdot*(m_f+m_add)/dt_es;
%     Fes=0;
else
    Fes=0;
end

dstatedt=  [xdot;
            (Frad+Fbear+Fdrag+Fwb+Fspring+Fgrav+Fhs+Fe+Fgen+Fes)/(m_f+m_add);
            -40*state(3)-50*state(4)+256*((m_f+m_add)*state(1)-state(6));
            8*state(3);
            F_known;
            state(5);
            -0.2*state(7)-0.08*state(8)+0.5*(-62.5*state(3)-78.12*state(4)+ ...
                 400*((m_f+m_add)*state(1)-state(6)));
            0.125*state(7);
            -Fgen*xdot-c1*(Fgen/Frated)^2-c2*abs(xdot/vrated)-c3;
%             -0.4*state(10)-0.54*state(11)+64*state(2);
%             0.5*state(10);
            -0.9*state(10)-0.85*state(11)-0.305*state(12)-0.18*state(13)+64*state(2);
            state(10);
            state(11);
            0.5*state(12);
            ];
```

B.2 Controllers

A number of MPC controllers have been implemented. Each uses two files identified by the suffix. The suffix can be defined on line 58 of *run_model.m*.

Suffix	Description
none	phase control by reference tracking (no constraints)
_c	phase control by reference tracking (incl. constraints)
_e	includes a noise model in the controller
_2	optimizes a very rough expression for the energy
_f	optimizes an accurate expression for the energy (no constraints)
_fc	optimizes an accurate expression for the energy (incl. hard constraints)
_fsc	optimizes an accurate expression for the energy (incl. hard and soft constraints)
_fsc_obs	optimizes an accurate expression for the energy (incl. constraints) but uses an observer to provide the velocity signal
_switch	uses the '_c' controller ordinarily but switches to '_fsc' to handle the constraints
_2step	optimizes an elaborate approximate expression for the energy by iterating from the reference tracking solution

The files are *formulate****.m* and *controller****.m*. The first defines the necessary parameters, the second is called by *Feeback.m* and calculates the control input. Another important file used by some of the controllers is *E_hat_cost_sc.m* which is a cost function representing the energy produced. It includes soft constraints.

This appendix only gives the code for the controllers which appear in this report. The CD delivered to the DCSC administration contains the following controllers:

All controllers make use of the standard model predictive control toolbox [1].

B.2.1 Reference Tracking Controller

Formulation

```
%
% Formulates the optimizition problem, the result of which governs the
% control law.
%
% In this case the constrained cost function:
%
%   J = [v_ref-v lambda*Fgen]'*[v_ref-v lambda*Fgen]
%
% is minimized over the prediction horizon. v_ref is an estimate of the
% optimal reference velocity and is a function of the excitation force.
%
% ouptut:
```

```
%
%   HH,AA,bb: define a quadratic optimization problem which solves:
%       minimize    mu'*HH*mu
%        mu         subject to AA*mu<bb*[1; x; v; 0; Fe_pred; v_ref];
%
%   vv: is a matrix such that the control input is calculated as:
%       Fgen = vv*[x; v; 0; Fe_pred(1); v_ref(1);...Fe_pred(N); v_ref(N); mu];
%
%   options2: defines the options used by the online optimizer quadprog
%   Gpredx: system to predict floater position

global vv HH AA bb options2 Gpredx;

% Define model of the system
mass=m_f+m_add;
A=[0 1; -ktot/mass -damp/mass]; B=[0; 1/mass;]; C=[0 1]; D=[0];
temp=c2d(ss(A,B,C,D),dt);
A=temp.a; B=temp.b; C=temp.c; D=temp.d;
Be=[0; 0];                  De=[1];
Bd=[B(:,1) zeros(2,1)];     Dd=[0 0];
Bu=B(:,1);                  Du=[0];

%position prediction
C2=[1 0];
B2=B(:,1);
[G,dim]=ss2syst(A,Be,B2,Bu,C2,De,Dd,Du);
[Gpredx,dim2]=pred(G,dim,N);

% Define constraints: -1<E*[x; v]<1
E=[1/(wb_limit) 0; 0 1/vrated];

% Define performance index:
%   z = Cz*[x; v] + Dze*e + Dzd*[Fe_pred; v_ref] + Dzu*Fgen
lambda=1e-6;
Cz=[0 -1; 0 0];      %performance: (v_ref-v lambda*Fgen)'
Dze=[0; 0];
Dzd=[0 1; 0 0];
Dzu=[0; lambda];

% Formulate optimization problem
[G,dim]=ss2syst(A,Be,Bd,Bu,C,De,Dd,Du,Cz,Dze,Dzd,Dzu);
[G,dim]=add_x(G,dim,E);
[G2,dim2]=pred(G,dim,N);
[G2,dim2]=add_u(G2,dim2,Frated);
[dGam]=dgamma(1,N,dim);
%[vv,HH,AA,bb]=contr(G2,dim2,dim,dGam);
[vv,HH,AA,bb]=contr_mod2(G2,dim2,dim,dGam);

options2=optimset('display','off');
```

Controller

```
function Fgen=controller_c(t,state,Fe_hat);
```

```
%
% Calculates value for control input
%
% In this case the constrained cost function:
%
%   J = [v_ref-v lambda*Fgen]*[v_ref-v lambda*Fgen]'
%
% is minimized over the prediction horizon. v_ref is an estimate of the
% optimal reference velocity and is a function of the excitation force.
%
% The controller first determines 'mu' by solving:
%
%       minimize    mu'*HH*mu
%        mu         subject to AA*mu<bb*[1; x; v; 0; Fe_pred; v_ref];
%
% Then the control force is calculated using:
%
%       Fgen = vv*[x; v; 0; Fe_pred(1); v_ref(1);...Fe_pred(N); v_ref(N); mu];

global N damp dt vv HH AA bb t2 Fgen options2 Frated
global vrated damp2 Gpredx wb_limit end_stops;

% Define prediction of disturbance input
vref=Fe_hat./(2*damp2);
dist=[];
for i=1:N
    if vref(i)>vrated
        vref(i)=vrated;
    end
    dist=[dist; Fe_hat(i); vref(i);];
end;

if length(t2)==1
    %initial value
    mu=quadprog(HH,[],AA,bb*[1; state(1); state(2); 0; dist],[],[],[],[],[],options2);
    Fgen=vv*[state(1); state(2); 0; dist; mu];
    Fgen=Fgen(1);
elseif length(t2)>1
    %if next step is reached
    if floor(t2(length(t2))/dt)>floor(t2(length(t2)-1)/dt)
        [mu,value,flag]=quadprog(HH,[],AA,bb*[1; state(1); state(2); 0; dist], ...
             [],[],[],[],[],options2);
        Fgen=vv*[state(1); state(2); 0; dist; mu];
        x_pred=(Gpredx.c)*[state(1); state(2)]+(Gpredx.d)*[0; Fe_hat; Fgen];
        Fgen=Fgen(1);

        if flag~=1
            text=sprintf(' * optimization issue in quadprog:
                           flag=%d, at t=%ds\n',flag,t2(length(t2)));
            disp(text);
            assignin('base','varInBase',text);
            evalin('base','message_out=sprintf(''%s%s'',message_out,varInBase);');
            assignin('base','varInBase',t2(length(t2)));
            evalin('base','t_error(end+1)=varInBase;');
```

```
            assignin('base','varInBase',flag);
            evalin('base','flag(end+1)=varInBase;');
            if Fgen<-Frated
                Fgen=-Frated;
            elseif Fgen>Frated
                Fgen=Frated;
            end;
        end;

    end;
end
assignin('base','varInBase',Fgen);
evalin('base','Fgen_out(end+1)=varInBase;');
```

B.2.2 Energy Maximizing Controller

Formulation

```
%
% Formulates the optimization problem, the result of which governs the
% control law.
%
% In this case the constrained non-linear cost function is minimized on-line.
% The cost function is found in E_hat_cost.m and minimizes:
%
%   J=v_pred*Fgen_pred+c1*(Fgen_pred/Frated)^2+c2*abs(v/v_rated)+P
%
% where P is a penalty which increases quadratically as the floater
% approaches the endstops.
%
% output:
%
%   Gpred, Gpredx: the systems used to predict the velocity and position
%   options2: the options used by the on-line optimizer 'fmincon'
%   Cc, ineq_matrix: prediction matricies such that
%           ( x_hat )  =  Cc * ( x0 )  + ineq_matrix * (Fgen_hat + Fe_hat)
%           ( v_hat )          ( v0 )
%   Constraints:
%           ( x_hat ) <= Constraints
%           ( v_hat )
%   x_max, v_max: maximum state values used to formulate penalty

global Gpred Gpredx options2 Cc ineq_matrix Constraints v_max x_max;

N=N+1;

% Define model of the system
mass=m_f+m_add;
A=[0 1; -ktot/mass -damp/mass]; B=[0; 1/mass;]; C=[0 1]; D=[0];
temp=c2d(ss(A,B,C,D),dt);
A=temp.a; B=temp.b; C=temp.c; D=temp.d;
Be=[0; 0];                  De=[1];
Bd=B(:,1);                  Dd=[0 0];
```

```
Bu=B(:,1);                  Du=[0];

% Define a linear velocity predictor
[G,dim]=ss2syst(A,Be,Bd,Bu,C,De,Dd,Du);
[Gpred,dim2]=pred(G,dim,N);
% Position prediction
C=[1 0];
[G,dim]=ss2syst(A,Be,Bd,Bu,C,De,Dd,Du);
[Gpredx,dim2]=pred(G,dim,N);

% Formulate constraints
Cc=[Gpredx.c; Gpred.c];
Ddxc=Gpredx.d; Ddxc=Ddxc(:,2:N+1);
Ddvc=Gpred.d; Ddvc=Ddvc(:,2:N+1);
ineq_matrix=[Ddxc; Ddvc];
x_max=end_stops;
v_max=vrated+0.5;
Constraints=[ones(N,1).*x_max; ones(N,1).*v_max];

options2=optimset('Display','off','TolFun',1e-1,'MaxIter',1e10,'MaxFunEvals',1e10, ...
     'Largescale','off');
```

Controller

```
function Fgen=controller_fsc(t,state,Fe_hat);
%
% Calculates value for control input for the constrained non-linear optimization
% problem:
%
%  J=sum_over_N ( v_pred*Fgen_pred+c1*(Fgen_pred/Frated)^2+c2*abs(v/v_rated) )
%
% where P is a penalty which increases quadratically as the floater
% approaches the endstops.
%
% The function E_hat_cost_sc expresses the predicted output power. It
% predicts the velocity as a function of the future generator and
% excitation forces by use of the linear model.

global damp N dt t2 Fgen Gpred c1 c2 c3 Frated vrated options2 Fgen0;
global Gpredx wb_limit end_stops v_max Cc ineq_matrix Constraints;

if length(t2)==1
   %initial control force
    Fgen0=zeros(N,1);
elseif length(t2)>1
    if floor(t2(length(t2))/dt)>floor(t2(length(t2)-1)/dt)
        % generator force constraints
        ub=ones(1,N)*Frated;
        ineq_vec=Constraints-Cc*[state(1); state(2)]-ineq_matrix*Fe_hat;

        % use fmincon to find optimal control force
        [Fgen,f,flag]=fmincon(@(Fgen) E_hat_cost_sc(Fgen,Fe_hat,Gpred,Gpredx, ...
           state(1),state(2),c1,c2,Frated,vrated,wb_limit),Fgen0,ineq_matrix, ...
```

```
           ineq_vec,[],[],[],[],[],options2);
        %[Fgen,f,flag]=patternsearch(@(Fgen) E_hat_cost(Fgen,Fe_hat,Gpred,state(1), ...
            state(2),c1,c2,Frated,vrated),Fgen0,[],[],[],[],-ub,ub,[],options2);
        Fgen0=Fgen;
        Fgen=Fgen(1);
        if Fgen>Frated
            Fgen=Frated;
        elseif Fgen<-Frated
            Fgen=-Frated;
        end

        % output flags
        assignin('base','varInBase',t2(length(t2)));
        evalin('base','t_error(end+1)=varInBase;');
        assignin('base','varInBase',flag);
        evalin('base','flag(end+1)=varInBase;');

        % if optimization terminates untimely give a message
        if flag<1
            text=sprintf(' * There were problems with fmincon:
                flag=%d at t=%ds\n',flag,t2(length(t2)));
            assignin('base','varInBase',text);
            evalin('base','message_out=sprintf(''%s%s'',message_out,varInBase);');
            x_pred=(Gpredx.c)*[state(1); state(2)]+(Gpredx.d)*[0; Fe_hat; Fgen0];
            v_pred=(Gpred.c)*[state(1); state(2)]+(Gpred.d)*[0; Fe_hat; Fgen0];
            if state(2)>0
                Fgen=-Frated;
            elseif state(2)<=0
                if state(1)<wb_limit
                    Fgen=Frated;
                else
                    Fgen=-Frated;
                end
            end;
        end;
    end;
end
```

Cost Function

```
function [E]=E_hat_cost_sc(Fgen_hat,Fe_hat,Gpred,Gpredx,x,v,c1,c2,Frated,vrated,wb_limit);
%
% Express the predicted negative energy output as a function of the predicted
% control force. The velocity is predicted as a function of the control force
% and excitation force using a linear model of the AWS. A penalty is added
% when the floater approaches the constraints.
%
% syntax: E=E_hat_cost(Fgen_hat,Fe_hat,Gpred,Gpredx,x,v,c1,c2,Frated,vrated,wb_limit);
%
%       Fgen_hat: predicted control force
%       Fe_hat: predicted excitation force
%       Gpred,Gpredx: prediction model
%       x,v: current floater position and velocity
```

```
%       c1,c2,Frated,vrated: generator parameters
%       wb_limit: position afterwhich water brakes activate

x_pred=(Gpredx.c)*[x; v]+(Gpredx.d)*[0; Fe_hat; Fgen_hat];
v_pred=(Gpred.c)*[x; v]+(Gpred.d)*[0; Fe_hat; Fgen_hat];

N=length(v_pred);
v_pred2=(v_pred(2:N)+v_pred(1:N-1))/2;

E=v_pred2'*Fgen_hat(1:N-1)+sum((c1/(Frated^2))*(Fgen_hat(1:N-1).^2)+(c2/vrated)*abs(v_pred(1

penalty_up=[ones(N,1).*wb_limit-0.5; ones(N,1).*vrated]-[x_pred(1:N); v_pred(1:N)];
penalty_down=[-ones(N,1).*wb_limit+0.5; -ones(N,1).*vrated]-[x_pred(1:N); v_pred(1:N)];
for i=1:2*N
    if penalty_up(i)>0
        penalty_up(i)=0;
    end;
    if penalty_down(i)<0
        penalty_down(i)=0;
    end;
end;

penalty_down=(penalty_down)'*(penalty_down)*4e6;
penalty_up=(penalty_up)'*(penalty_up)*4e6;
penalty_Fgen=0;

E=E+penalty_down+penalty_up+penalty_Fgen;
```

B.2.3 Switching Controller

Formulation

```
%
% *** Switcher Formulation ***
%
% Formulates the optimization problem, the result of which governs the
% control law.
%
% In this case the controller switches between an unconstrained reference
% tracker and minimizing a soft constrained non-linear cost function when
% the motion approaches the constraints.
%
% output:
%
%   Gpred,Gpredx: the system used to predict the velocity and position
%   options2: the options used by the on-line optimizer 'fmincon'
%   Cc, ineq_matrix: prediction matricies such that
%           ( x_hat )  =  Cc * ( x0 )  + ineq_matrix * (Fgen_hat + Fe_hat)
%           ( v_hat )          ( v0 )
%   Constraints:
%           ( x_hat ) <= Constraints
%           ( v_hat )
%   vv,HH,AA,bb: matrices used to define reference tracker control signal
```

```
global Gpred Gpredx options2 Cc ineq_matrix Constraints vv HH AA bb;

% Define model of the system
mass=m_f+m_add;
A=[0 1; -ktot/mass -damp/mass]; B=[0; 1/mass;]; C=[1 0]; D=[0];
temp=c2d(ss(A,B,C,D),dt);
A=temp.a; B=temp.b; C=temp.c; D=temp.d;

% First define the unconstrained reference tracking problem
Be=[0; 0];                  De=[1];
Bd=[B(:,1) zeros(2,1)];     Dd=[0 0];
Bu=B(:,1);                  Du=[0];

% Define constraints: -1<E*[x; v]<1
E=[1/(wb_limit) 0; 0 1/vrated];

%   - define performance index for reference tracking
lambda=1e-6;
Cz=[0 -1; 0 0];      %performance: (v_ref-v lambda*Fgen)'
Dze=[0; 0];
Dzd=[0 1; 0 0];
Dzu=[0; lambda];     %weighting on input

%   - formulate feedback law
[G,dim]=ss2syst(A,Be,Bd,Bu,C,De,Dd,Du,Cz,Dze,Dzd,Dzu);
[G,dim]=add_x(G,dim,E);
[G2,dim2]=pred(G,dim,N);
[G2,dim2]=add_u(G2,dim2,Frated);
[dGam]=dgamma(1,N,dim);
[vv,HH,AA,bb]=contr_mod2(G2,dim2,dim,dGam);

% Next define matrices needed for non-linear optimization problem:
ktot2=1*ktot;
damp=damp;
A=[0 1; -ktot2/mass -damp/mass]; B=[0; 1/mass;]; C=[1 0]; D=[0];
temp=c2d(ss(A,B,C,D),dt);
A=temp.a; B=temp.b; C=temp.c; D=temp.d;

Be=[0; 0];                  De=[1];
Bd=B(:,1);                  Dd=[0 0];
Bu=B(:,1);                  Du=[0];

%   - define a linear position predictor
N2=round(N/4);
C=[1 0];
[G,dim]=ss2syst(A,Be,Bd,Bu,C,De,Dd,Du);
[Gpredx,dim2]=pred(G,dim,N2);
%   - define a linear velocity predictor
C=[0 1];
[G,dim]=ss2syst(A,Be,Bd,Bu,C,De,Dd,Du);
[Gpred,dim2]=pred(G,dim,N2);

%   - formulate constraints
```

```
Cc=[Gpredx.c; Gpred.c];
Ddxc=Gpredx.d; Ddxc=Ddxc(:,2:N2+1);
Ddvc=Gpred.d; Ddvc=Ddvc(:,2:N2+1);
ineq_matrix=[Ddxc; Ddvc];
x_max=end_stops;
v_max=vrated;
Constraints=[ones(N2,1).*x_max; ones(N2,1).*v_max];

options2=optimset('Display','off','TolFun',1e-2,'MaxIter',1e10,'MaxFunEvals',1e10, ...
   'Largescale','off');
```

Controller

```
function Fgen=controller_switch(t,state,Fe_hat);
%
% Controller which combines the soft-constrained energy maximizing controller
% with the unconstrained reference tracking controller.
%
% The soft-constrained energy maximizing controller only switches on when
% the predicted trajectory crosses the constraints. It switches back when
% the prediction has moved well away from the constraints and a time
% interval has passed.

global damp dt N t2 Fgen Gpred c1 c2 c3 Frated vrated options2 Cc ineq_matrix
global vv HH AA bb damp2 Constraints
global Gpredx wb_limit end_stops
persistent t_switch v_switch sw_flag Fgen0 t_on;

% if sw_flag==[] then set sw_flag=0;
if sw_flag<inf
else
    sw_flag=0;
end;
if t_on<inf
else
    t_on=0;
end;

if length(t2)==1
   %initial control force
   Fgen0=-Fe_hat;
elseif length(t2)>1
    if floor(t2(length(t2))/dt)>floor(t2(length(t2)-1)/dt)
        % the following determines whether to use the reference tracker or
        % the energy maximizer:
        %   sw_flag=0: use reference tracker
        %   sw_flag=1: use energy maximizer
        N2=length(Gpredx.c);
        x_pred=(Gpredx.c)*[state(1); state(2)]+(Gpredx.d)*[0; Fe_hat(1:N2); Fgen0(1:N2)];
        if sw_flag==0
            %if prediction is too large switch to energy maximizer
            if max(x_pred)>wb_limit
                sw_flag=1;
```

```
        t_on=t2(length(t2));
    end;
end;
if sw_flag==1
    %if prediction small enough again and the energy maximizing controller
    %has been on for long enough switch to reference tracker
    if t2(length(t2))-t_on>2
    if max(x_pred)<wb_limit-2
        sw_flag=0;
    end;
    end;
end;

% use the non-linear optimization controller
if sw_flag==1
    ub=ones(1,N2)*Frated;
    ineq_vec=Constraints-Cc*[state(1); state(2)]-ineq_matrix*Fe_hat(1:N2);

    % use fmincon to find optimal control force
    [Fgen,f,flag]=fmincon(@(Fgen) E_hat_cost_sc(Fgen,Fe_hat(1:N2),Gpred, ...
       Gpredx,state(1),state(2),c1,c2,Frated,vrated,wb_limit),Fgen0(1:N2), ...
       ineq_matrix,ineq_vec,[],[],-ub,ub,[],options2);
    Fgen0=Fgen;
    Fgen=Fgen(1);

    % output some useful stuff
    assignin('base','varInBase',t2(length(t2)));
    evalin('base','t_error(end+1)=varInBase;');
    assignin('base','varInBase',1);
    evalin('base','swit(end+1)=varInBase;');
    assignin('base','varInBase',flag);
    evalin('base','flag(end+1)=varInBase;');

    % if optimization terminates untimely give a message
    if flag<1
        text=sprintf(' * There were problems with fmincon: flag=%d at
           t=%ds\n',flag,t2(length(t2)));
        assignin('base','varInBase',text);
        evalin('base','message_out=sprintf(''%s%s'',message_out,varInBase);');
        Fgen=-Frated;
    end;
end;

% use uncon. ref. tracker
if sw_flag==0
% Define prediction of disturbance input
vref=Fe_hat./(2*damp2);
if abs(vref(1)-state(2))>1
    t_switch=t2(length(t2));
    v_switch=state(2);
end
%this smoothens out the transition incase the actual velocity and
%reference velocity differ by too much
if t2(length(t2))-t_switch<1
```

```
            vref=vref+(v_switch-vref).*exp(1*(t_switch-t2(length(t2))-[0:dt:N*dt-dt]'));
        end;
        dist=[];
        for i=1:N
            dist=[dist; Fe_hat(i); vref(i);];
        end;

        %solve quadprog
        [mu,value,flag]=quadprog(HH,[],AA,bb*[1; state(1); state(2); 0; dist], ...
           [],[],[],[],[],options2);
        Fgen0=vv*[state(1); state(2); 0; dist; mu];
        Fgen=Fgen0(1);

        %output some useful data
        assignin('base','varInBase',0);
        evalin('base','swit(end+1)=varInBase;');

        %incase the optimization procedure is not successful use
        %Fgen=Frated and output a message
        if flag~=1
            text=sprintf(' * optimization issue in quadprog: flag=%d,
                       at t=%ds\n',flag,t2(length(t2)));
            disp(text);
            assignin('base','varInBase',text);
            evalin('base','message_out=sprintf(''%s%s'',message_out,varInBase);');
            assignin('base','varInBase',t2(length(t2)));
            evalin('base','t_error(end+1)=varInBase;');
            assignin('base','varInBase',flag);
            evalin('base','flag(end+1)=varInBase;');
            if Fgen<-Frated
                Fgen=-Frated;
            elseif Fgen>Frated
                Fgen=Frated;
            end;
        end;
        end;
    end;
end
```

B.2.4 2-step Controller

Formulation

```
%
% 2-step formulation
% Formulates the optimizition propblem, the result of which governs the
% control law.
%
% The controller first finds the solution to the reference tracking
% problem, then uses the predicted trajectory and predicted generator force
% to optimize and expression for the change in energy (see report).
%
% outputs:
```

```
%       vv,HH,AA,bb: matrices for quadprog to solve refernec tracking
%       Cy,Dyd,Dyu,L,K: matrices used to define change in predicted E as a
%           function of the generator force
%       Gpredx,Gpredv: systems to predict floater position and velocity
%       constraints: maximum change in velocity and generator force
%       ub: vector containing upper bound in control force
%       options2,options3: options for quadprog and fmincon repsectively

global vv HH AA bb options2 Cy Dyd Dyu L K Gpredv Gpredx constraints ub options3;

% Define model of the system
mass=m_f+m_add;
A=[0 1; -ktot/mass -damp/mass]; B=[0; 1/mass;]; C=[0 1]; D=[0];
temp=c2d(ss(A,B,C,D),dt);
A=temp.a; B=temp.b; C=temp.c; D=temp.d;
Be=[0; 0];                  De=[1];
Bd=[B(:,1) zeros(2,1)];     Dd=[0 0];
Bu=B(:,1);                  Du=[0];

%constraints
E=[1/(wb_limit) 0; 0 1/vrated];

% Formulate reference tracking optimization problem
% Define performance index
Cz=[0 -1; 0 0];      %performance: (v_ref-v lambda*Fgen)'
Dze=[0; 0];
Dzd=[0 1; 0 0];
Dzu=[0; 1e-6];     %weighting on input
[G,dim]=ss2syst(A,Be,Bd,Bu,C,De,Dd,Du,Cz,Dze,Dzd,Dzu);
[G2,dim2]=pred(G,dim,N);
[dGam]=dgamma(1,N,dim);
[vv1,HH,AA,bb]=contr_mod2(G2,dim2,dim,dGam);%   - define performance index for ref. track
lambda=1e-6;
Cz=[0 -1; 0 0];      %performance: (v_ref-v lambda*Fgen)'
Dze=[0; 0];
Dzd=[0 1; 0 0];
Dzu=[0; lambda];     %weighting on input

%   - formulate feedback law
[G,dim]=ss2syst(A,Be,Bd,Bu,C,De,Dd,Du,Cz,Dze,Dzd,Dzu);
[G,dim]=add_x(G,dim,E);
[G2,dim2]=pred(G,dim,N);
[G2,dim2]=add_u(G2,dim2,Frated);
[dGam]=dgamma(1,N,dim);
[vv,HH,AA,bb]=contr_mod2(G2,dim2,dim,dGam);
options2=optimset('Display','off','LargeScale','off');

order=length(A);
% Velocity prediction
Be=[0; 0];                  De=[1];
Bd=[B(:,1)];                Dd=[0];
Bu=B(:,1);                  Du=[0];
[G,dim]=ss2syst(A,Be,Bd,Bu,C,De,Dd,Du);
[Gpredv,dim2]=pred(G,dim,N);
```

```
% Position prediction
C=[1 0];
[G,dim]=ss2syst(A,Be,Bd,Bu,C,De,Dd,Du);
[Gpredx,dim2]=pred(G,dim,N);

A=Gpredv.a;
B=Gpredv.b;
C=Gpredv.c;
D=Gpredv.d;
D=D(:,2:N+1);

K=c1/Frated^2;
L=c2/vrated;

%define the system:
%   [dv; dF]=Cyd*[x; v]+Dyd*[v0; F0; Fe_hat]+Dyu*Fgen_hat
Cy=[C; zeros(N,2)];
Dyd=[-eye(N) zeros(N) D; [zeros(N) -eye(N) zeros(N)]];
Dyu=[D; eye(N)];

%new constraints
dv_max=0.5;
dF_max=1e4;
ub=Frated;
constraints=[ones(N,1).*dv_max; ones(N,1).*dF_max];

options3=optimset('Display','off','LargeScale','off');
```

Controller

```
function Fgen=controller_2step(t,state,Fe_hat);
%
% 2 step controller
%
% First finds the solution to the reference tracking problem, then uses the
% predicted trajectory and predicted generator force to optimize and
% expression for the change in energy (see report).

global N dt t2 Fgen Gpredx Gpredv Frated vrated c1 c2;
global vv HH AA bb options2 options3;
global damp K L Cy Dyd Dyu constraints ub;

if length(t2)==1
    Fgen=0;
elseif length(t2)>1
    %if next step is reached
    if floor(t2(length(t2))/dt)>floor(t2(length(t2)-1)/dt)
        %first compute reference tracking control input signal
        %   -define prediction of disturbance input
        vref=Fe_hat./(2*damp);
        dist=[];
        for i=1:N
            dist=[dist; Fe_hat(i); vref(i);];
```

```
        end;

        %   -solve quadprog
        [mu,value,flag]=quadprog(HH,[],AA,bb*[1; state(1); state(2); 0; dist], ...
           [],[],[],[],[],options2);
        F0=vv*[state(1); state(2); 0; dist; mu];
        v0=(Gpredv.c)*[state(1); state(2)]+(Gpredv.d)*[0; Fe_hat; F0];

        %next find the optimum of the approximated increase in energy
        %   -constrain changes in v and Fgen
        ineq_mat=[Dyu; -Dyu];
        ineq_vec=[constraints; constraints]-[Cy; -Cy]*[state(1); state(2)]- ...
            [Dyd; -Dyd]*[v0; F0; Fe_hat];

        f=(F0+L*10)'*Dyu(1:N,:)+(v0+K)'*Dyu(N+1:2*N,:);
        [Fgen_hat,fun,flag]=linprog(f',ineq_mat,ineq_vec,[],[],-ub*ones(N,1), ...
            ub*ones(N,1),F0,options3);
        y=Cy*[state(1); state(2)]+Dyd*[v0; F0; Fe_hat]+Dyu*Fgen_hat;
        dv=y(1:N);
        dF=y(N+1:2*N);

        if flag~=1
            disp('bleh');
            Fgen=F0(1);
        else
            Fgen_hat';
            Fgen=Fgen_hat(1);
        end
        if Fgen<-Frated
            Fgen=-Frated;
        elseif Fgen>Frated
            Fgen=Frated;
        end;
        assignin('base','varInBase',flag);
        evalin('base','flag(end+1)=varInBase;');
    end
end
```

B.3 Calculating the hydrodynamic loads

The hydrodynamic loads acting on the AWS can be derived from the sea surface profile. The hydrodynamic loads are the loads acting on the outside of the floater due to the waves passing over the AWS. The loads calculated are the vertical force (the excitation force F_e) and the resulting horizontal force profile, characterized by the horizontal force acting on the bottom of the floater (F_x), and the horizontal moment acting about the bottom of the floater (M_x). The derivation is covered in section 3.2 of the literature study report.

```
%
% This m-file computes the external hydrodynamic forces acting on the AWS
```

```
% from the sea surface profile. The computation is based on linear wave
% theory as discussed in chapter 3.2 of the literature study report.
%
% Output:
%   t  = time
%   Fe = excitation force
%   Fx = horizontal force
%   Mx = horizontal moment around the bottom of the floater

fs=5;
global dt;
dt=1/fs;
tfinal=250;
t=0:dt:tfinal;
N=length(t);
f=linspace(0,fs,N);
om=2*pi*f;

v=spline(time(:,n)-time(1,n),v(:,n),t);
A=fft(v)./N;

gain=S_F*rho_sea*g;

%find k as a function of omega
ktemp=0.001:0.001:10;
temp=tanh(ktemp.*h);
temp2=g.*ktemp;
for l=1:length(om)
    [m,idx]=min(abs(om(l)^2-temp2.*temp));
    k(l)=ktemp(idx);
end;
% clear k;
% k=(om(1:500).^2)/g;
% k=[k ones(1,N-500).*k(500)];

Kc=sin(k*d_out/2)./(k*d_out/2);

decay_top=cosh(k.*(h-(h-h_f/2-x_0)))./cosh(k.*h).*Kc;
decay_bot=cosh(k.*(h-(h+h_f/2-x_0)))./cosh(k.*h).*Kc;
for i=ceil(N/2):N
    decay_top(i)=decay_top(2*ceil(N/2)-i);
    decay_bot(i)=decay_bot(2*ceil(N/2)-i);
end

fE=gain.*A.*decay_top-(S_F-S_f)*rho_sea*g.*A.*decay_bot;

Fe=real(ifft(fE)).*N;

%**** horizontal loads ****
u=[];
a=[];
dz=2;
for z=0:dz:h_f
    d=h-x_0+h_f/2-z;
```

```
    Kh=cosh(k.*(h-d))./sinh(k.*h);
    U=om.*Kh;
    Acc=-j*om.*om.*Kh;
    for i=ceil(N/2):N
        U(i)=U(2*ceil(N/2)-i);
        Acc(i)=Acc(2*ceil(N/2)-i);
    end
    u=[u; ifft(U.*A).*N];
    a=[a; ifft(Acc.*A).*N];
end;

for i=1:length(t)
    dfdx=real(a(:,i)).*rho_sea*pi*d_out*dz*C_M+ ...
          0.5*rho_sea*d_out*dz*C_D.*real(u(:,i)).*abs(real(u(:,i)));
    dfdxF=fit([0:dz:h_f]',dfdx,'linearinterp');
    dmdx=fit([0:dz:h_f]',dfdx.*[0:dz:h_f]','linearinterp');
    F_hor(i)=integrate(dfdxF,h_f,0);
    M_hor(i)=integrate(dmdx,h_f,0);
end;

global Fx Mx;
Fx=[F_hor'];
Mx=[M_hor'];
```

Appendix C

Approximating the Radiation Force

As the floater of the AWS oscillates it radiates a wave. The radiation force is the reaction of the radiated wave onto the floater. This appendix briefly summarizes the results obtained from linear wave theory, and discusses modelling it.

C.1 An Analytical Expression for the Raditaion Force

Using linear wave theory the radiation force is given as:

$$F_{rad} = -m_{add}\ddot{z} - \int_0^t 2K(t-\tau)\dot{z}(\tau)d\tau \tag{C.1}$$

where m_{add} represents the added inertia to the floater and $K(t)$ represents the memory of the fluid. The extra inertia (m_{add}) is a constant and can be determined numerically or experimentally. The memory term ($K(t)$) can be determined from the Haskind relation:

$$K(\omega) = \frac{\omega k(\omega)|H_{\eta\rightarrow F_e}(\omega)|^2}{4\pi\rho g^2 D(k(\omega)h)} \tag{C.2}$$

where $H_{\eta\rightarrow F_e}(\omega)$ is the transfer function from the wave elevation to the excitation force as derived in chapter 3.2 of the literature study report, $k(\omega)$ is the wave number, and $D(k(\omega)h)$ is the depth function and is given as:

$$D(k(\omega)h) = \left(1 + \frac{2k(\omega)h}{\sinh(2k(\omega)h}\right)\tanh(k(\omega)h) \tag{C.3}$$

Figure C.1 shows the resulting transfer function. Note that the transfer function is purely real.

The literature study report and Falnes [6] give a detailed derivation.

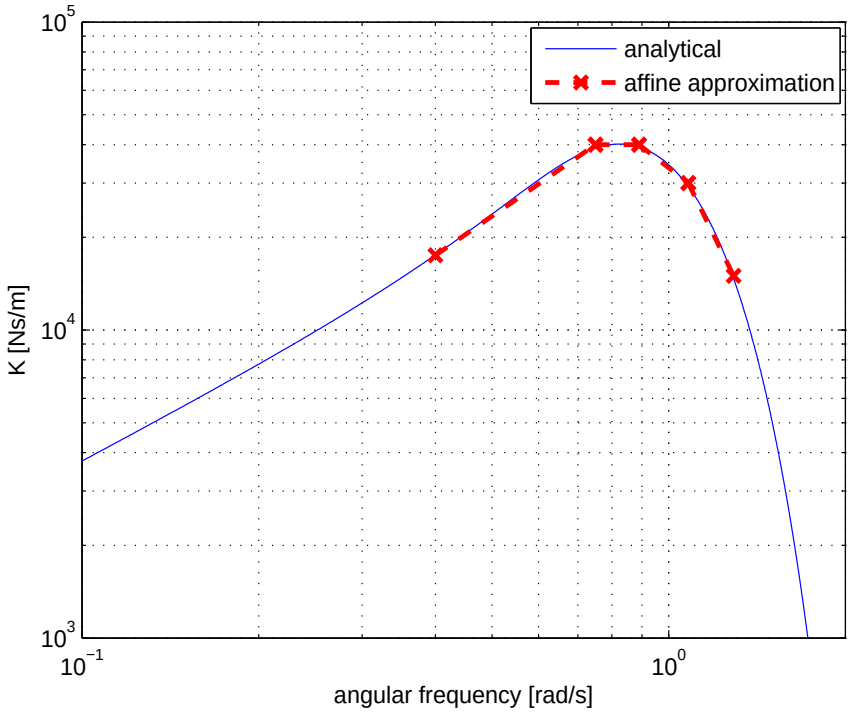

Figure C.1: Schematic of the AWS

C.2 Approximating the Memory Term as a Damping

The radiation force can be expressed as a damping coefficient multiplied by the floater velocity, where the magnitude of the coefficient depends on the frequency of the floater motion. Assuming the frequency of floater motion is similar to that of the excitation force (which is true for the optimal motion) then the frequency of the floater motion is centered on the tuning frequency of the AWS (i.e. the average frequency equals the tuning frequency). Hence the average radiation damping coefficent can be related to the tuning frequency of the AWS.

$K(\omega)$ is approximated as a set of affine functions as shown in figure C.1. The tuning frequency is then used to calculate the average radiation damping coefficient. This provides a rough expression for the radiation force.

C.3 Approximating the Memory Term as a Linear System

It is also possible to approximate the radiation force as a linear system with the same frequency response as $K(\omega)$. One way to do this is illustrated in [8], whereby the impulse repsonse of the memory term is mimicked by a sum of exponential functions:

$$K(t) \approx \sum_{i=1}^{N} \alpha_i e^{\beta_i t} \tag{C.4}$$

The memory term can then be approximated as:

$$\int_0^t K(t-\tau)\dot{z}(\tau)d\tau \approx \sum_{i=1}^{N} \int_0^t \alpha_i e^{\beta_i t}\dot{z}d\tau \tag{C.5}$$

Each integral $I_i = \int_0^t \alpha_i e^{\beta_i t}\dot{z}d\tau$ can be expressed by the second order linear system:

$$\begin{pmatrix} \dot{\Re(I_i)} \\ \dot{\Im(I_i)} \end{pmatrix} = \begin{pmatrix} \Re(\beta_i) & -\Im(\beta_i) \\ \Re(\beta_i) & \Im(\beta_i) \end{pmatrix} \begin{pmatrix} \Re(I_i) \\ \Im(I_i) \end{pmatrix} + \begin{pmatrix} \Re(\alpha_i) \\ \Im(\alpha_i) \end{pmatrix} \dot{z} \tag{C.6}$$

where $\Re(\cdot)$ and $\Im(\cdot)$ represent the real and imaginary parts of their arguments respectively.

Finally the memory term can be expressed by the following linear system:

$$\begin{pmatrix} \dot{\Re(I_1)} \\ \dot{\Im(I_1)} \\ \vdots \\ \dot{\Re(I_N)} \\ \dot{\Im(I_N)} \end{pmatrix} = \begin{pmatrix} \Re(\beta_1) & -\Im(\beta_1) & \cdots & 0 & 0 \\ \Re(\beta_1) & \Im(\beta_1) & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & \Re(\beta_N) & -\Im(\beta_N) \\ 0 & 0 & \cdots & \Re(\beta_N) & \Im(\beta_N) \end{pmatrix} \begin{pmatrix} \Re(I_1) \\ \Im(I_1) \\ \vdots \\ \Re(I_N) \\ \Im(I_N) \end{pmatrix} + \begin{pmatrix} \Re(\alpha_1) \\ \Im(\alpha_1) \\ \vdots \\ \Re(\alpha_N) \\ \Im(\alpha_N) \end{pmatrix} \dot{z}$$

$$K(t) \approx \begin{pmatrix} 1 & 0 & \cdots & 1 & 0 \end{pmatrix} \begin{pmatrix} \Re(I_1) \\ \Im(I_1) \\ \vdots \\ \Re(I_N) \\ \Im(I_N) \end{pmatrix} \tag{C.7}$$

Appendix D

Model Predictive Control

This appendix gives a short introduction to Model Predictive Control and derives some of the tools used in the main report. Most of the appendix is taken from reference [1]. Reference [7] also provides a good course on model predictive control.

D.1 Basic Principle

MPC is a control methodology which, as the name implies, makes use of a model to predict future states. At any point in time the current state of the system can be considered as the initial conditions. It is then possible, using a model of the system, to express an estimate of the future trajectory of the states in terms of the input. The future state values can then be used to express a peformance index or cost function for which an optimal can be found. For example, the most apparent performance index for the case of the AWS would be the net power produced by the generator. Including the system constraints the problem comes down to a constrained optimization problem. The controller solves this problem and yields a future control input from the initial state until some finite time into the future. The last future time for which the optimal input is predicted is called the horizon.

Next the start of the computed optimal input is implemented. The system responds but inevitably does not respond in exactly the same way as predicted, due to reality-model mismatch and unexpected or unpredictable disturbances. The system output then provides the controller with new initial conditions and the controller makes another estimate of the future optimal control input. Figure D.1 shows a flowchart illustrating the principle of the MPC controller

D.2 Using Linear Models to make Predictions

When the dynamics of a system can be approximated by a linear system it is a simple matter to predict the state of the system in terms of the initial state and the future input and disturbance. The system can be written as a discrete-time linear

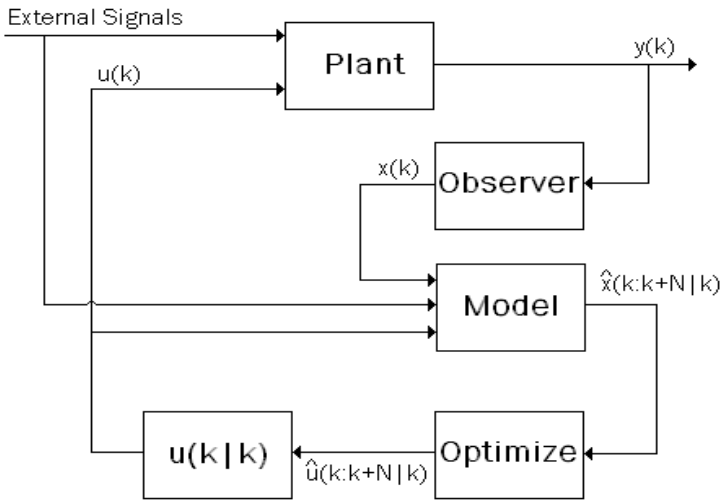

Figure D.1: Basic Principle of MPC

system:

$$\begin{aligned}
x[k+1] &= Ax[k] + Bu[k] \\
x[k+2] &= Ax[k+1] + Bu[k+1] \\
&= A\left(Ax[k] + Bu[k]\right) + Bu[k+1] \\
&= A^2x[k] + \begin{pmatrix} AB & B \end{pmatrix}\left(u[k] \;\; u[k+1]\right)^T \\
x[k+3] &= A^3x[k] + \begin{pmatrix} A^2B & AB & B \end{pmatrix}\left(u[k] \;\; u[k+1] \;\; u[k+2]\right)^T \\
&\ldots \\
x[k+N] &= A^Nx[k] + \begin{pmatrix} A^{N-1}B & A^{N-2}B & \ldots & B \end{pmatrix}\left(u[k] \;\; u[k+1] \;\; \ldots \;\; u[k+N]\right)^T
\end{aligned} \tag{D.1}$$

where $x[k]$ is the state vector and $u[k]$ is a vector containing the inputs to the open loop (the control signal, and the disturbances) at time step k.

When a signal is expressed as a linear function of the state (e.g. an output signal (y) or a performance signal (z)) it is a trivial matter to predict it aswell.

$$\begin{aligned}
y[k] &= C_yx[k] + D_yu[k] \\
y[k+1] &= C_yx[k+1] + D_yu[k+1] \\
&= C_yAx[k] + C_yBu[k] + D_yu[k+1] \\
y[k+2] &= C_yx[k+2] + D_yu[k+2] \\
&= C_yA^2x[k] + C_y\begin{pmatrix} AB & B \end{pmatrix}\left(u[k] \;\; u[k+1]\right)^T + D_yu[k+2] \\
&\ldots
\end{aligned} \tag{D.2}$$

It is convenient to express the resulting system as:

$$\begin{aligned}
\hat{y} &= \tilde{C}_yx[k] + \tilde{D}_{yd}\hat{d} + \tilde{D}_{yu}\hat{u} \\
\hat{z} &= \tilde{C}_zx[k] + \tilde{D}_{yd}\hat{d} + \tilde{D}_{zu}\hat{u}
\end{aligned} \tag{D.3}$$

where the decoration $\hat{}$ denotes that the variable is a vector containing predicted values. For example $\hat{y} = (y[k] \;\; y[k+1] \;\; ... \;\; y[k+N])^T$. The inputs are split into d and u, the external signals and the control signal respectively.

D.3 Reformulating to the Quadratic Programming Problem

A quadratic programming problem is an optimization problem of the form:

$$\min_{\vec{x}} \;\; \frac{1}{2}\vec{x}^T H \vec{x} + \vec{x}^T \vec{f} \tag{D.4}$$

where H is a constant matrix and $\vec{f}$ is a constant vector. Reference [5] provides an overview of the programming problem.

Very quick and reliable algorithms exist to find the constrained optimum of the quadratic programming problem, as long as the matrix H is positive definite. Hence it is desirable to express the optimization problem used to determine the optimal control signal as such. The optimization problem becomes a quadratic programming problem when the controller tries to minimize a cost function of the form:

$$J = \min_{\hat{u}} \;\; \sum_{i=0}^{N} z[k+i|k]^2 \tag{D.5}$$

where z is a linear combination of the state variables. This is shown below.

Rewriting equation D.5 we get:

$$J = \min_{\hat{u}} \;\; \hat{z}^T \hat{z} \tag{D.6}$$

Recall that $\hat{z}$ is given as:

$$\hat{z} = \tilde{C}_z x[k] + \tilde{D}_{yd}\hat{d} + \tilde{D}_{zu}\hat{u} \tag{D.7}$$

By substitution:

$$\begin{aligned} J &= \min_{\hat{u}} \;\; \left(\tilde{C}_z x[k] + \tilde{D}_{yd}\hat{d} + \tilde{D}_{zu}\hat{u}\right)^T \left(\tilde{C}_z x[k] + \tilde{D}_{yd}\hat{d} + \tilde{D}_{zu}\hat{u}\right) \\ &= \min_{\hat{u}} \;\; \hat{u}^T \tilde{D}_{zu}^T \tilde{D}_{zu}\hat{u} + 2\hat{u}^T \tilde{D}_{zu}^T \left(\tilde{C}_z x[k] + \tilde{D}_{yd}\hat{d}\right) + \left(\tilde{C}_z x[k] + \tilde{D}_{yd}\hat{d}\right)^T \left(\tilde{C}_z x[k] + \tilde{D}_{yd}\hat{d}\right) \end{aligned} \tag{D.8}$$

Next we define:

$$\begin{aligned} H &\equiv 2\tilde{D}_{zu}^T \tilde{D}_{zu} \\ \vec{f} &\equiv 2\tilde{D}_{zu}^T \left(\tilde{C}_z x[k] + \tilde{D}_{yd}\hat{d}\right) \\ K &\equiv \left(\tilde{C}_z x[k] + \tilde{D}_{yd}\hat{d}\right)^T \left(\tilde{C}_z x[k] + \tilde{D}_{yd}\hat{d}\right) \end{aligned} \tag{D.9}$$

and substitute:

$$\begin{aligned} J &= \min_{\hat{u}} \ \frac{1}{2}\hat{u}^T H \hat{u} + \hat{u}^T \vec{f} + K \\ &= \min_{\hat{u}} \ \frac{1}{2}\hat{u}^T H \hat{u} + \hat{u}^T \vec{f} \end{aligned} \tag{D.10}$$

It can be shown that $H = 2\tilde{D}_{zu}^T \tilde{D}_{zu}$ is always positive definite. Hence, as long as the performance index (z) can be expressed as a linear combination of the state variables, it is possible to write the controller optimization problem as a quadratic programming problem.

Appendix E

Random Search Results

This appendix presents the results of the random search used to investigate the properties of the nonlinear optimization problem. The optimization problem is defined as:

$$\min_{\hat{F}_{gen}} \quad (-E) = \min_{\hat{F}_{gen}} \quad \sum_{i=1}^{N} \left(\dot{z}[i+0.5]\hat{F}_{gen}[i] + c_1 \left(\frac{\hat{F}_{gen}[i]}{F_{rated}} \right)^2 + c_2 \left| \frac{\dot{z}[i]}{v_{rated}} \right| \right) \tag{E.1}$$

where the velocity ($\dot{z}$) is a function of the excitation force, the generator force, and the initial state. The generator parameters (c_1, c_2, v_{rated} and F_{rated}) also play and important role in the properties of the cost function. The values used are the same as those used by the controller and are listed in table E.

The optimization problem is solved in for 12 different cases, each with different excitation force signals and different initial conditions. In each case the optimization problem is solved 200 times. Figure E shows the resulting range of control forces found, where the dashed lines show the excitation force. Table E gives the average cost values found and the standard deviation.

When the excitation force is very low there is a large variation in the resulting control signals. The variation in the resulting cost values is also particularly large. In such sea states the controller is likely to perform poorly. It could be worth it to implement a multi-start in the controller, but the energy provided by the sea is small anyway. For larger excitation forces the range in control forces is very small and the variation in the cost value is also small.

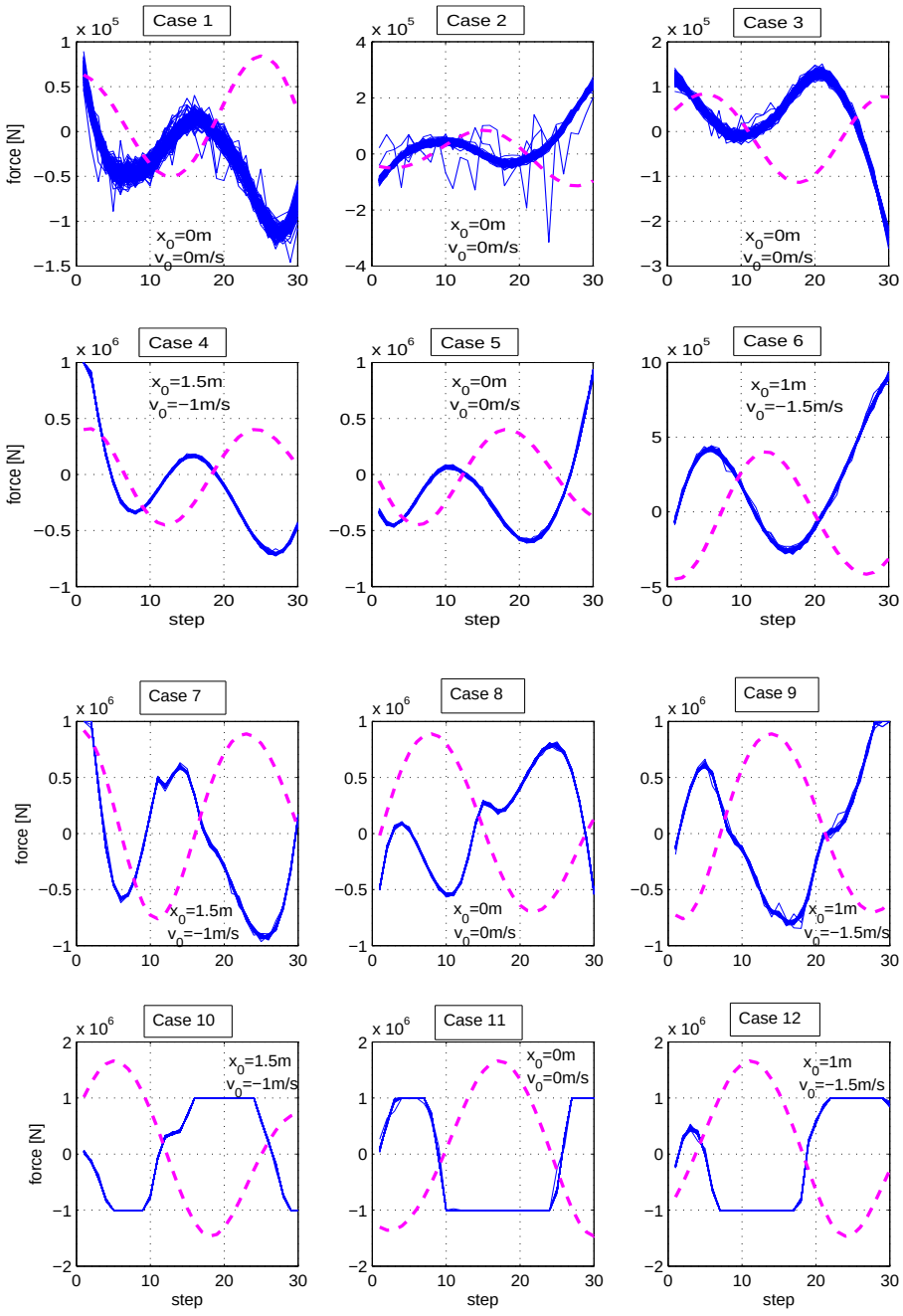

Figure E.1: Random search results: generator forces

Parameter	Value
c_1	250 kW
c_2	20 kW
F_{rated}	1 MN
v_{rated}	2.2 m/s

Table E.1: Generator parameters

case number	mean cost value	standard deviation	ratio (std/mean)
1	-6.58E+04	579	0.88%
2	-1.51E+05	10700	7.09%
3	-4.07E+05	407	0.10%
4	-4.90E+06	409	0.01%
5	-3.96E+06	593	0.01%
6	-7.68E+07	732	0.00%
7	-1.28E+07	1410	0.01%
8	-9.37E+06	297	0.00%
9	-1.82E+07	3190	0.02%
10	-1.98E+07	532	0.00%
11	-1.95E+07	10200	0.05%
12	-2.55E+07	942	0.00%

Table E.2: Cost values

Bibliography

[1] van den Boom, T. J. J., Backx, T. C. P. M., *Model Predictive Control: Lecture Notes (SC4060)*, Delft University of Technology, Delft, Netherlands, 2007

[2] Yavuz, H., McCabe, A., Aggidis, G., Widden, M. B., *Calculation of the Performance of Resonant Wave Energy Converters in Real Seas*, University of Lancaster, UK, 2006

[3] Babarit, A., Duclos, G., Clément, A., H., *Comparison of Latching Control strategies for a Heaving Wave Energy Device in Random Sea*, Applied Ocean Research, Volume 26, Issue 5, July 2004, Pages 227-238

[4] Prado, M. G. S., Gardner, F., Damen, M., Polinder, H., *Modelling and Test Results of the Archimedes Wave Swing*, Teamwork Technology, Zijdewind, Netherlands, 2006

[5] van den Boom, T. J. J., De Schutter, B., *Optimization in Systems and Control: Lecture Notes (SC4090)*, Delft University of Technology, Delft, Netherlands, 2004

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

[7] Maciejowski, J. M., *Predictive Control: with Constraints*, Pearson Education Limited[3], London, UK, 2002

[8] Babarit, A., Clément, A., H.,*Optimal Latching Control of a Wave Energy Device in Regular and Irregular Waves*, Applied Ocean Research, Volume 28, Issue 2, April 2006, Pages 77-91

[9] Korde, U., A., *On Providing a Reaction for Efficient Wave Energy Absorption by Floating Devices*, Applied Ocean Research, Volume 21, Issue 5, October 1999, Pages 235-248

[10] Eidsmoen, H., *Tight-moored Amplitude-limited Heaving-buoy Wave-energy Converter with Phase Control*, Applied Ocean Research, Volume 20, Issue 3, June 1998, Pages 157-161

[11] Oelering, G. J. *Ocean Wave Modelling using White-Noise Filtering Techniques for use in Offshore Systems Analysis: Graduate Thesis*, Delft University of Technology, Delft, Netherlands, 1994

[12] Mandal, S., Witz, J. A., Lyons, G. J. *Reduced Order ARMA Spectral Estimation of Ocean Waves*, Applied Ocean Research, Volume 14, Issue 5, 1992, Pages 303-312