

# Modeling and Design Analysis of a Permanent Magnet Linear Synchronous Generator

Technical Report UIUC-ESDL-2013-01\*

Xin Niu<sup>†</sup>  
Engineering System Design Lab  
University of Illinois at Urbana-Champaign

August 25, 2013

## Abstract

Ocean wave energy conversion is a popular research area due to the increasing scarceness of nonrenewable energy resources. However, extracting wave energy is not a simple process. For example, the design and construction of wave energy take-off devices that are reliable, long-lasting, and efficient is a non-trivial endeavor. In this report, the physics-based theories behind a permanent magnet linear generator are reviewed. A model of the generator is built from a series of independent and dependent parameters. The model is simulated using MATLAB<sup>®</sup> with a specific set of parameter values. Two of the independent parameters are studied and improvement potential is revealed.

---

\*This technical report was submitted in partial fulfillment of an independent study project at the University of Illinois at Urbana-Champaign Department of Industrial and Enterprise Systems Engineering.

<sup>†</sup>B.S. Candidate in Industrial Engineering, Department of Industrial and Enterprise Systems Engineering, University of Illinois at Urbana-Champaign, [xinniu2@illinois.edu](mailto:xinniu2@illinois.edu).  
©2013 Xin Niu

# Contents

<b>1</b>	<b>Permanent Magnet Linear Synchronous Generators</b>	<b>3</b>
1.1	Electromagnetic Theory . . . . .	4
<b>2</b>	<b>Dynamic System Model</b>	<b>7</b>
2.1	System Description . . . . .	8
2.2	Generator Model . . . . .	9
2.3	Wave Model . . . . .	15
<b>3</b>	<b>Numerical Studies</b>	<b>16</b>
3.1	Simulation Results . . . . .	16
3.2	Design Parameter Studies . . . . .	26
3.3	Model Stiffness Study . . . . .	32
<b>4</b>	<b>Conclusion</b>	<b>36</b>
<b>A</b>	<b>Appendix: Modeling Code</b>	<b>39</b>
<b>B</b>	<b>Appendix: Engineering Science Elements in this Report</b>	<b>48</b>

# 1 Permanent Magnet Linear Synchronous Generators

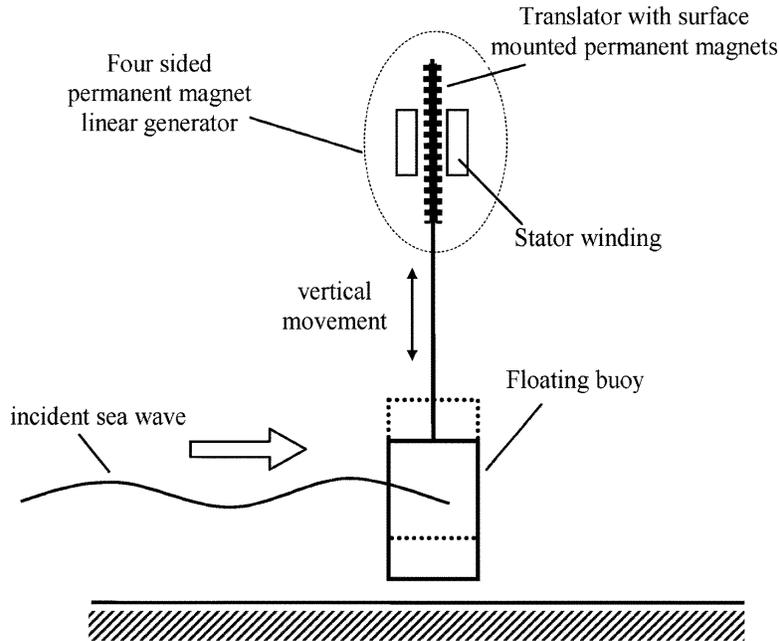


Figure 1: Heaving Linear Generator Power Take-Off Device for Wave Energy Conversion (from Ref. [1]).

This report describes a model for a permanent magnet (PM) linear synchronous generator that is part of a system for converting ocean wave energy into electricity [2]. A dynamic mathematical model of such a generator was developed based on Ref. [2], and tested using a variety of simulations. A generator is known as “synchronous” when the waveform of the generated voltage is synchronized with the rotation of the generator. Permanent magnets are used to provide the field excitation in a linear synchronous generator. Due to mechanical constraints, linear motors have large and variable airgaps [3]. Therefore, the magnetic circuit has a large reluctance. Magnetic reluctance, or magnetic resistance, is a property of magnetic circuits. The concept of magnetic reluctance is analogous to resistance in an electrical circuit. However, instead of dissipating electrical energy, magnetic reluctance stores magnetic energy. An electric field causes an electric current to follow the path of least resistance. Similarly, a magnetic field causes magnetic flux to follow the path of least magnetic reluctance. A permanent magnet linear synchronous motor requires magnets with a large coercive force, which is the intensity of the applied magnetic field required to reduce the magnetization of that material to zero after the magnetization of the material has been driven to saturation. When a material has a large coercive

force it is difficult for the material to lose its magnetization. Using magnets with a large coercive force results in a permanent magnet generator that has stable properties over time. Ferrite and rare-earth magnets are examples of permanent magnets with a large coercive force [3].

## 1.1 Electromagnetic Theory

Pertinent elements of electromagnetic theory will be reviewed here as a foundation for understanding the operation of permanent magnet linear synchronous generators. For a circuit carrying a current  $I$  as shown in Fig. 2, the magnetic field intensity  $H$  at a point  $P$  is defined by the following equation:

$$\mathbf{H} = \frac{1}{4\pi} I \int_C \frac{d\mathbf{l} \times \mathbf{r}_1}{r^2}. \quad (1.1)$$

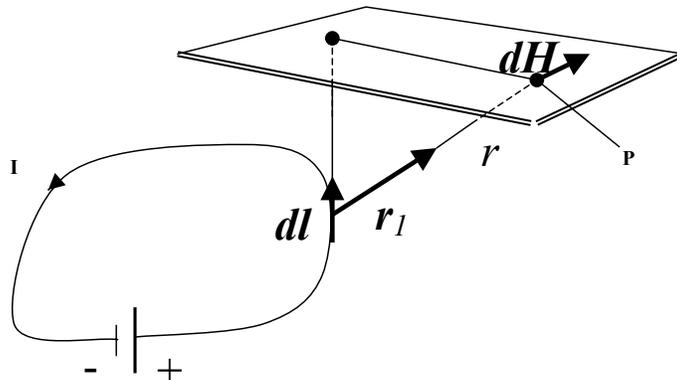


Figure 2: The Magnetic Intensity  $d\mathbf{H}$  Produced by a Line Element  $d\mathbf{l}$  of the Current  $I$  in a Circuit (from Ref. [4]).

The integration is carried out over a closed circuit  $C$ . The unit vector  $\mathbf{r}_1$  and the distance  $r$  show the direction and distance respectively from the source to the point of observation. Magnetic field intensity is expressed in amperes per meter (A/m).

A magnetic field  $B$  is quantified by the force that it exerts on a conductor carrying an electrical current. It is related with the magnetic field intensity as follows:

$$\mathbf{B} = \mu\mathbf{H}, \quad (1.2)$$

where the constant  $\mu$  is the permeability of the medium the magnetic field passes through. Magnetic field is expressed in Tesla (T or  $\text{N} \cdot \text{A}^{-1} \cdot \text{m}^{-1}$ ). If the material

is nonlinear, the permeability is a function of  $\mathbf{B}$ :

$$\mu = \frac{\mathbf{B}}{H(\mathbf{B})}. \quad (1.3)$$

The permeability of free space  $\mu_0$  is defined as:  $\mu_0 = 4\pi \times 10^{-7} \text{ T} \cdot \text{m/A}$ .

The magnetic flux through a surface  $S$  is the component of the magnetic field  $\mathbf{B}$  passing through that surface:

$$\Phi = \int_S \mathbf{B} \cdot d\mathbf{A}, \quad (1.4)$$

and is measured in Weber (W or  $\text{T} \cdot \text{m}^2$ ).

The magnetization  $\mathbf{M}$  is the magnetic moment per unit volume at a given point in a medium. The magnetic moment is associated with the orbital and spinning motion of electrons. It has the same unit as the magnetic field intensity, A/m. When magnetization  $\mathbf{M}$  is present, it combines with the magnetic field intensity  $\mathbf{H}$  to produce the magnetic induction through the field equation:

$$\mathbf{B} = \mu_0(\mathbf{H} + \mathbf{M}). \quad (1.5)$$

The fundamentals of a permanent magnet linear synchronous generator are closely related to magnetic field and Maxwell equations. Three of the four equations are related to permanent magnet linear synchronous generators by Ampere's Law:

$$\nabla \times \mathbf{H} = \mathbf{J} + \frac{\partial \mathbf{D}}{\partial t}. \quad (1.6)$$

Here  $\mathbf{J}$  is the volume current density in the circuit. The second term ( $\partial \mathbf{D} / \partial t$ ) involves electric displacement  $\mathbf{D}$ , which is a vector field that accounts for the effects of free charge within materials. When the cyclic variation of  $\mathbf{D}$  has low frequency, which is the case in this report, the second term is considerably smaller than  $\mathbf{J}$ . Therefore, electric displacement will be neglected in this report. With this simplification, Eqn. (1.6) can be rewritten as follows:

$$\oint_C \mathbf{H} \cdot d\mathbf{l} = \int_S \mathbf{J} \cdot d\mathbf{A}. \quad (1.7)$$

Equation (1.7) states that line integral of  $\mathbf{H}$  over a closed curve  $C$  is equal to the current crossing the surface  $S$  bounded by  $C$ . Often the same current crosses the surface bounded by the curve  $C$  several times. A solenoid is a coil wound into a tightly packed helix, as illustrated in Fig. 3. A current runs through the coil and creates a magnetic field. With a solenoid,  $C$  could follow the axis and then return outside the solenoid. The total current crossing the surface is then the current in each turn multiplied by the number of turns. Equation (1.7) describes how the magnetic field intensity is determined by the distribution of current in a circuit. This equation governs the behavior of the stator part of a

permanent magnet linear synchronous generator. The stator is the stationary portion of the generator. It contains an electric circuit, and current is induced in the stator circuit by the moving permanent magnets located on the translator. The stator and translator are depicted in Fig. 1.

$$\nabla \times \mathbf{B} = 0. \quad (1.8)$$

Equation (1.8) is the second of the three Maxwell's equations discussed here, and is also known as Gauss law. It states that the net flux of  $\mathbf{B}$  in any volume is zero. Unlike electrical field lines, a magnetic field line must complete a closed, continuous curve. This must always be taken in to account in the design of the magnetic circuit of a generator.

$$\nabla \times \mathbf{E} = -\frac{\partial \mathbf{B}}{\partial t}. \quad (1.9)$$

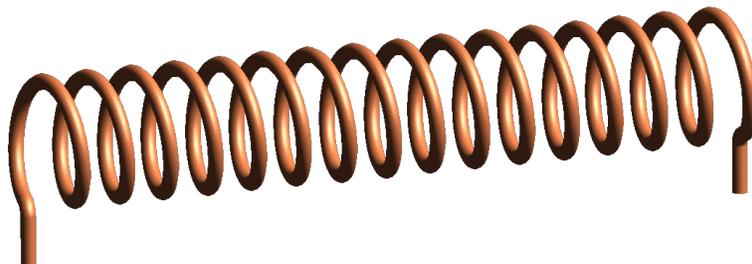


Figure 3: An Example of a Solenoid.

Equation (1.9) is the third of Maxwell's equations relevant to linear generators, and states that the curl of the electric field is equal to the negative time derivative of the magnetic field. This demonstrates the duality of electric and magnetic fields. In physics, in the static case, electromagnetism has two separate facets: electric fields and magnetic fields. This gives rise to the electromagnetic dual concept. Expressions of either of these will have a dual expression of the other. The reason behind this is related to special relativity when using Lorentz transformation to transform electric fields to magnetic fields. Some examples of the duality of electric and magnetic fields include electric field and magnetic field, electric displacement field and magnetizing field and Faraday's law and Ampere's law. Equation (1.9) can be rewritten by using Stokes Theorem on the left hand side and Eqn. (1.4) on the right hand side:

$$e = -\frac{d\Phi}{dt}, \quad (1.10)$$

where  $e$  is the electromotive force (EMF) and  $\Phi$  is the magnetic flux through a closed surface.

Equation (1.10) is the most widespread formulation of Faraday's Law, which states: The induced electromotive force, which is the voltage generated, in any

closed circuit is equal to the negative of the time rate of change of the magnetic flux through the circuit. The direction of the current induced in the circuit is such that its magnetic field opposes, to a greater or lesser extent depending on the resistance of the circuit, the change in flux. If the closed circuit comprises  $N$  turns close together, each intercepting the same magnetic flux, then the electromotive forces add up, resulting in an  $N$  times larger electromotance. With this in mind, we can define  $N\Phi$  as the flux linkage  $\Lambda$ :

$$\Lambda = N\Phi, \tag{1.11}$$

and Faraday's Law for circuits with multiple turns can be rewritten as:

$$e = -\frac{d\Lambda}{dt}. \tag{1.12}$$

## 2 Dynamic System Model

There are two primary types of permanent magnet linear synchronous generators: tubular permanent magnet linear synchronous generators and flat permanent magnet linear synchronous generators. The tubular permanent magnet linear synchronous generator is cylindrical. The force-to-weight ratio of such machines has been proved to be higher than flat generators [5]. However, flat generators are less expensive and easier to build.

In this report two models of a permanent magnet linear synchronous generator will be built and simulated. A series of differential equations will be formulated to describe the incident ocean wave and the reaction of the permanent magnet linear synchronous generator. The incoming ocean wave is assumed to be sinusoidal. For the first model, we ignore the reaction force applied by the generator. The second model includes the effect of the reaction force of the generator. These two models were then used to simulate dynamic generator behavior. Generator performance was evaluated using each model.

## 2.1 System Description

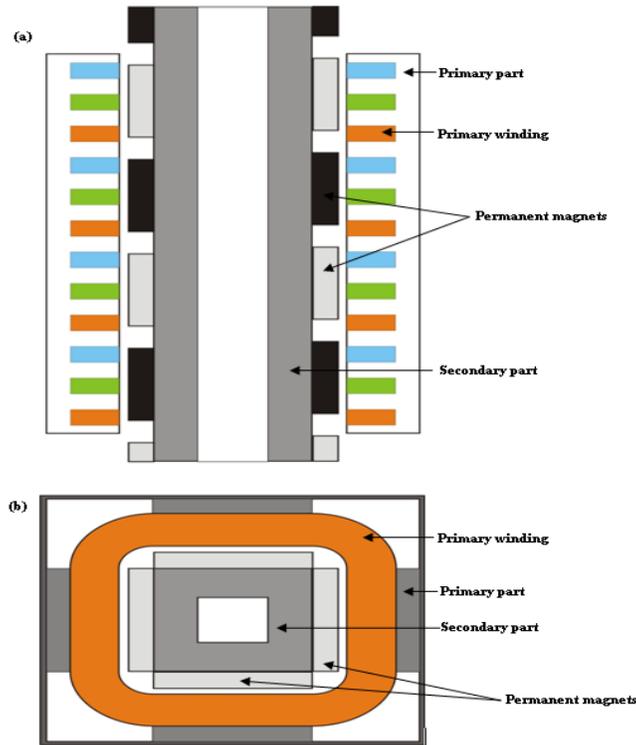


Figure 4: Schematic of the Quasi-Flat Tubular Permanent Magnet Linear Synchronous Generator: (a) Longitudinal Cross-Section (b) Top-View (from Ref. [2]).

The proposed structure of the permanent magnet linear synchronous generator is shown schematically in Fig. 4, which is based on the structure from Ref. [2]. It consists of four flat primary elements and four secondary elements enclosed within one housing. The secondary elements, which correspond to the translator in Fig. 1, move with the buoy and are called translators, which are the non-stationary parts in a generator. The primary elements, which correspond to the stator winding in Fig. 1, don't move and are called stators. Each of the flat sides is similar in structure to a flat linear machine as shown in Fig. 5.

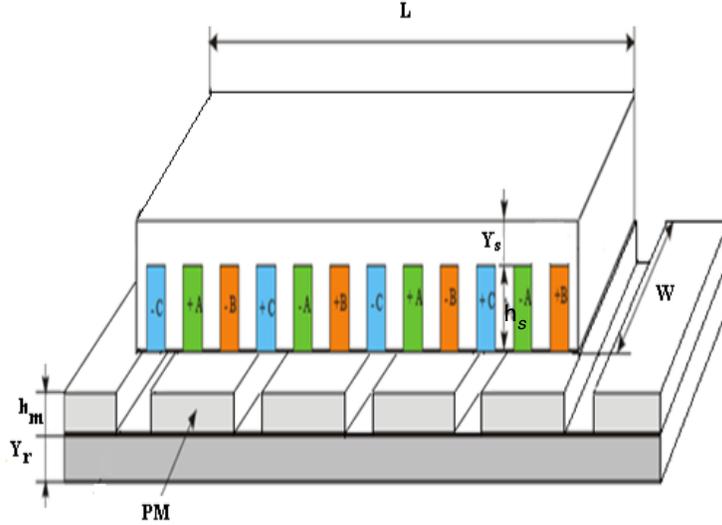


Figure 5: Structure of the Flat Permanent Magnet Linear Synchronous Generator (from Ref. [2]).

Figure 5 depicts the stator, which is the top element in the figure, and the translator, which is the bottom element shown in the figure. The stator contains the windings. Normally a permanent magnet linear synchronous generator contains a number of individual windings stacked together. The model in this report describes a permanent magnet linear synchronous generator with three phases. The three phases of windings are labeled A, B, and C.  $L$  is the length of the stator core.  $Y_s$  is the stator yoke thickness.  $h_s$  is the height of the slot.  $W$  is the width of the stator.  $h_m$  is the height of the magnet.  $Y_r$  is the translator yoke thickness.

The translator will move along with the buoy that moves with the ocean wave. Since the magnets on the translator are also moving with it, the magnetic field that passes through the motionless stator windings will change with time. According to Faraday's Law discussed in the previous section, this will induce currents in the circuit. This is the mechanism that transforms the kinetic energy of the ocean wave into electricity. A dynamic model of this energy conversion based on differential equations will be presented later in this section.

## 2.2 Generator Model

The structure of the primary core is shown as below, where  $g_{eq}$  is the equivalent air gap. The primary core is the stator. The secondary core is the translator.



Figure 6: Schematic Representation of Unslotted Primary Core (from Ref. [2]).

The stator circuit windings run through slots in the stator. The slot dimensions are shown below:

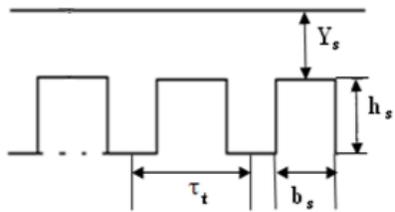


Figure 7: Illustration of Primary Core Slots (from Ref. [2]).

An equivalent circuit model of the generator is shown in Fig. 8:

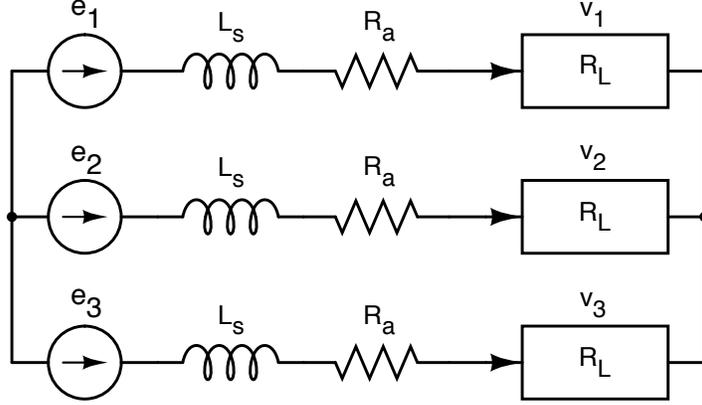


Figure 8: Equivalent Circuit of 3-Phase Winding of Generator (from Ref. [2]).

Table 1: Generator Parameters (from Ref. [2]).

Name	Symbol	Value	Units
Number of phases	$m$	3	
Number of poles	$p$	6	
Number of slots/pole/phase	$q$	1	
Number of armatures	$M_s$	4	
Number of turns per coil	$N$	82	
Length of the stator core	$L$	0.432	m
Stator width	$W_s$	0.2	m
Air gap	$g_a$	0.002	m
Height of magnets	$h_m$	0.006	m
Permissible flux density in translator core	$B_r$	1.2	T
Coercive magnetic field intensity	$H_c$	905000	A/m
Armature resistance	$R_a$	1.5	$\Omega$
Load resistance	$R_L$	7.5	$\Omega$
Synchronous inductance	$L_s$	0.115	H

Table 1 lists model parameters. The values used for most parameters listed in this table are based on the values given in Ref. [2]. Some parameters, namely  $m$ ,  $p$ ,  $q$ ,  $M_s$ ,  $N$ ,  $L$ ,  $W_s$ ,  $g_a$  and  $h_m$ , are physical parameters and independent from each other. They are variables that can be changed by designers, and are treated as independent design variables in this report. However, it should be noted that they still have finite ranges of validity due to physical constraints. In addition, not all combinations of variable values within allowed ranges are physically meaningful.  $B_r$  and  $H_c$  are properties of the magnetic material used in Ref. [2].

We assume the magnetic material choice is fixed, so these two parameter values are not changeable here. Please note that some parameters, including  $R_a$ ,  $R_L$  and  $L_s$ , are dependent parameters that will change with the physical design of the generator, but are considered to be fixed here for the specific generator design detailed in Ref. [2]. This assumption simplifies modeling, but does reduce model fidelity to some degree.

The number of turns per coil  $N$  refers to the number of turns in each of the three stator windings. The model in this report is a four-sided permanent magnet linear synchronous generator, with the windings on the stator elements. For the generator architecture used here, the stator elements are known also as armatures, which the power-producing component in a generator. Therefore this generator has four armatures, one for each side.

The armature resistance  $R_a$  is the resistance of each phase of the armature circuits. The synchronous inductance  $L_s$  is the inductance in each phase, and the load resistance  $R_L$  is the resistance of the load in the circuit. The load refers to an external electrical system being powered by the generator. These elements are depicted in the equivalent circuit model that is illustrated in Fig. 8.

The pole pitch  $\tau_p$  is the overall width of each pole. It may be expressed as a function of stator core length  $L$  and the number of poles  $p$ :

$$\tau_p = \frac{L}{p}. \quad (2.1)$$

The tooth pitch  $\tau_t$  may be derived from the pole pitch, and is illustrated in Fig. 7:

$$\tau_t = \frac{\tau_p}{mq}, \quad (2.2)$$

where  $m$  is the number of phases and  $q$  is the number of slots per phase per pole. Here we assume the slot width is:

$$b_s = 0.016 \text{ m}. \quad (2.3)$$

Knowing  $b_s$ , we can calculate tooth width:

$$b_t = \tau_t - b_s. \quad (2.4)$$

In a slotted armature, the effective area of the magnetic flux path is substantially reduced due to its geometry. This results in an increase in air gap reluctance. This phenomenon may be understood by studying Carter's coefficient ( $K_c$ ) for air gap reluctance. In this simple method for estimating the air gap reluctance, we assume that the air gap flux is uniformly distributed over the whole of the slot pitch except for a fraction of slot width. This is possible because of how small the air gap is relative to the rest of the geometry (i.e., the stator is very close to the magnets). In fact, the magnetic field just above the magnet surface is perpendicular to the surface and uniform. Therefore, we can assume that the stator element is in this nearly uniform magnetic field.

Carter proposed a model for estimating air gap reluctance based on a coefficient that depends upon the ratio of slot width to gap length [6, 7]. In this approach we account for increased air gap reluctance by computing an equivalent air gap  $g_{eq}$ , which then influences flux density calculations. The equivalent air gap is calculated as follows:

$$g_{eq} = K_c \cdot g_a, \quad (2.5)$$

where Carter's Coefficient obtained by evaluating the following expression [2]:

$$K_c = \frac{\tau_t(5g_a + b_s)}{\tau_t(5g_a + b_s) - b_s^2}. \quad (2.6)$$

For each phase, the air-gap flux density (accounting for increased reluctance) is:

$$\phi = \frac{B_r h_m H_c \mu_0}{h_m H_c \mu_0 - g_{eq} B_r}. \quad (2.7)$$

The voltage equations associated with the equivalent circuit in Fig. 8 are as follows, where  $e_{1,2,3}$  are the values of induced electromotive force for each phase:

$$e_1 = R_a i_1 + L_s \frac{di_1}{dt} + v_1, \quad (2.8)$$

$$e_2 = R_a i_2 + L_s \frac{di_2}{dt} + v_2, \quad (2.9)$$

$$e_3 = R_a i_3 + L_s \frac{di_3}{dt} + v_3, \quad (2.10)$$

where  $v_{1,2,3}$  are generator terminal phase voltages (computed using Eqns. (2.16) below), and  $i_{1,2,3}$  are the phase currents. The voltage induced in the phase winding by the translator motion is:

$$e_{ph} = K_E \cos\left(\frac{\pi}{\tau} z\right) v(t), \quad (2.11)$$

where  $z$  is the displacement of the buoy,  $v(t)$  is the vertical velocity of the buoy and the voltage constant  $K_E$  is calculated using:

$$K_E = M_s \cdot W_s \cdot N_{ph} \cdot \phi \cdot v_{av}, \quad (2.12)$$

where  $v_{av}$  is the average buoy speed defined as  $v_{av} = \frac{2}{\pi} v_m$  with  $v_m$  being the vertical speed amplitude, and  $N_{ph} = N_c \cdot p \cdot q$  is the number of turns per coil for each phase. The induced voltages in the windings for each phase are computed as follows:

$$e_1 = K_E \cos\left(\frac{\pi}{\tau} z\right) v(t), \quad (2.13)$$

$$e_2 = K_E \cos\left(\frac{\pi}{\tau} z - \frac{2\pi}{3}\right) v(t), \quad (2.14)$$

$$e_3 = K_E \cos\left(\frac{\pi}{\tau} z - \frac{4\pi}{3}\right) v(t). \quad (2.15)$$

The generator terminal phase voltages are:

$$v_j = i_j R_L, \quad j = 1, 2, 3. \quad (2.16)$$

The electromagnetic force for each phase (ph = 1, 2, 3) can therefore be computed using:

$$F_{\text{ph}} = N_{\text{ph}} \phi_{\text{ph}} \frac{\pi}{\tau} \sin\left(\frac{\pi z}{\tau}\right) i_{\text{ph}}. \quad (2.17)$$

We will consider two model fidelity levels here. The lower fidelity model, referred to here as the *simple* model, ignores the electromagnetic force in the formulation of the governing differential equations. This means that only the induced current in the circuit is considered without the effect it has on the moving translator. In this model, due to the absence of the electromagnetic force,  $v(t)$  will be the same as the wave elevation equation that defines ocean wave elevation as a function of time  $u(t)$  (see Eqn. (2.23)). This will be known as the *simple model*, and the corresponding system of governing differential equations is:

$$\frac{d}{dt} i_1 = \frac{e_1 - v_1 - i_1 R_a}{L_s}, \quad (2.18)$$

$$\frac{d}{dt} i_2 = \frac{e_2 - v_2 - i_2 R_a}{L_s}, \quad (2.19)$$

$$\frac{d}{dt} i_3 = \frac{e_3 - v_3 - i_3 R_a}{L_s}, \quad (2.20)$$

where the phase currents  $(i_1, i_2, i_3)$ . In this model we assume that the buoy (translator) position track the wave elevation exactly (see Section 2.3 for a discussion of the ocean wave model). This is not what occurs in reality, as the difference between wave elevation and buoy position is what provides the buoyancy force that moves the translator to produce electrical energy. This simplification, however, allows investigation of PTO dynamics via a simple third-order dynamic model. A detailed implementation of this model is presented in Appendix A.

The second, higher-fidelity, model accounts for the electromagnetic force and the mechanical dynamics of the buoy. This model is referred to here as the *complex* model. Inclusion of the wave energy converter (WEC) translator dynamics results in a fifth-order dynamic model. Buoy dynamics depend on mass and geometry. Therefore,  $v(t)$  will no longer be the same as  $u(t)$  since electromagnetic force is not ignored. Instead, it will be defined by the electromagnetic dynamics of the whole system. There are several possible shapes of the buoy, including cylindrical and spherical. Cylindrical buoys are easy to simulate because buoy cross section does not change with respect to vertical position [8, 9]; a cylindrical cross section is used in this model.

The two state variables added to the complex model include the translator position ( $z$ ) and the translator velocity ( $v$ , which is equivalent to buoy velocity). In this model buoy position is decoupled from ocean wave elevation. The

following two equations are added to Eqns. (2.18)-(2.20) to form a set of five differential equations:

$$\frac{d}{dt}z = v, \quad (2.21)$$

$$\frac{d}{dt}v = -\frac{[R_r(T) + R_v] \left[ v - \frac{2\pi}{T} A \cos\left(\frac{2\pi}{T}t\right) \right] + \rho g \pi a^2 \left[ z - A \sin\left(\frac{2\pi}{T}t\right) \right]}{M + m_r(\infty)}. \quad (2.22)$$

The parameters in Eqns. (2.21) and (2.22) are defined in Section 2.3.

### 2.3 Wave Model

To understand and quantify the performance of the permanent magnet linear synchronous generator we need a mathematical model of the incoming ocean wave. In both the simple and complex models the incident ocean wave is assumed to be sinusoidal, and in the simple model we assume that the ocean wave elevation is unaffected by the WEC. These are strong assumptions. In reality, ocean waves are irregular. An irregular incident wave field can be modeled as a linear superposition of a finite number of regular, or sinusoidal, wave components [8, 10]. This model is not identical to the real situation, but this simplification allows us to understand how the general model will work. We can understand, through this simplified model, how the PTO will transform the kinetic energy in the incoming ocean wave into electricity.

In the simple model the wave speed amplitude ( $u_m$ ), which is the maximum vertical speed of the wave, is assumed to be 2.2 m/s, and the period of the wave ( $T$ ) is assumed to be 12.6 s. With the speed amplitude and period specified, we can now define the wave elevation equation:

$$u(t) = u_m \sin\left(\frac{2\pi t}{T}\right), \quad (2.23)$$

which describes buoy speed as a function of time.

In the complex model, however, the existence of the electromagnetic force complicates the dynamics of the buoy. In the complex model the displacement amplitude of the wave is assumed to be  $A = 4$  m. The period  $T$  is still assumed to be 12.6 s. While in the simple model we assume that the buoy and translator position is coupled with the wave elevation, the complex model allows for relative motion between the wave and buoy. This allows the generation of net buoyancy force, and requires that we account for the dynamics of the buoy and several other effects. First, we need to consider the volume and mass of the buoy. The mass and displaced volume determine the buoyant force exerted on the buoy by the water. We can calculate the displaced volume based on the buoy radius  $a$  and the position of the buoy relative to the wave elevation. The draft  $b$  is the depth of the bottom surface of the buoy in still water.

The mass of the buoy is assumed to be  $M = 1000$  kg. When the buoy is moving in the water, it will entrain (move) some amount of water with it. This phenomenon gives rise to the term *added mass*, which is the mass of the water

that moves along with the buoy as it oscillates in the water. When formulating a dynamic model of the buoy system the mass used is the sum of the buoy mass and the added mass. According to the calculation in Ref. [8], the added mass  $m_r(\infty)$  for the buoy in this report is 289 kg. Since the buoy is moving relative to the water, the radiation resistance  $R_r$ , which depends on the geometry of the buoy and the wave frequency [11], will have an effect on the motion of the buoy.  $R_r$  is calculated to be 16 Ns/m [8]. In addition, the viscous damping  $R_v$  will also play a role in the dynamics, which is calculated to be 717 Ns/m [8]. All the necessary parameters are in Table 2. After taking all these effects into consideration we have the system described as Eq. (2.21) and Eq. (2.22).

Table 2: Heaving Cylinder WEC Parameters for Complex Model [8, 9].

Name	Symbol	Value	Units
Radius	$a$	0.5	m
Draft	$b$	1	m
Water depth	$h$	10	m
Mass	$M$	1000	kg
Density of water	$\rho$	1025	kg/m <sup>3</sup>
Added mass	$m_r(\infty)$	289	kg
Radiation resistance damping	$R_r(T)$	16	Ns/m
Viscous damping	$R_v$	717	Ns/m

## 3 Numerical Studies

### 3.1 Simulation Results

We simulated the model using MATLAB. We used the solver `ode45`—a variable step Runge-Kutta method—to solve differential equations numerically. The solver `ode45` is appropriate for nonstiff<sup>1</sup> problems when a medium order of accuracy is desired.

#### Simple Model Results

First we simulated the simpler model, which does not include the electromagnetic force. While the simulation spanned 350 seconds, the time period of primary interest in this study is between 250 s–350 s. After 250 s have elapsed the system has had enough time to reach steady state operating conditions.

As illustrated in Fig. 9 and Fig. 10,  $z$  and  $v$  both have a sinusoidal shape. The position  $z$  is always positive, while the velocity  $v$  is symmetric about 0. The maximum and minimum values of  $z$  are 8.8235 m and 0, respectively. The velocity  $v$  has a maximum of 2.2 m/s and a minimum of -2.2 m/s, as specified

<sup>1</sup>Model stiffness is addressed in Sec. 3.3.

in the previous section. Figure 10 is scaled such that exactly one period of steady-state conditions is displayed.

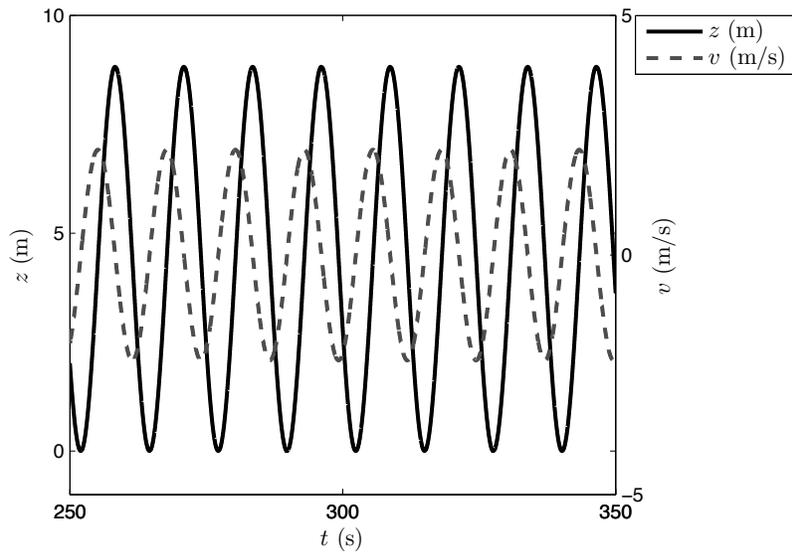


Figure 9: Simple Model Results for Position and Velocity vs. Time.

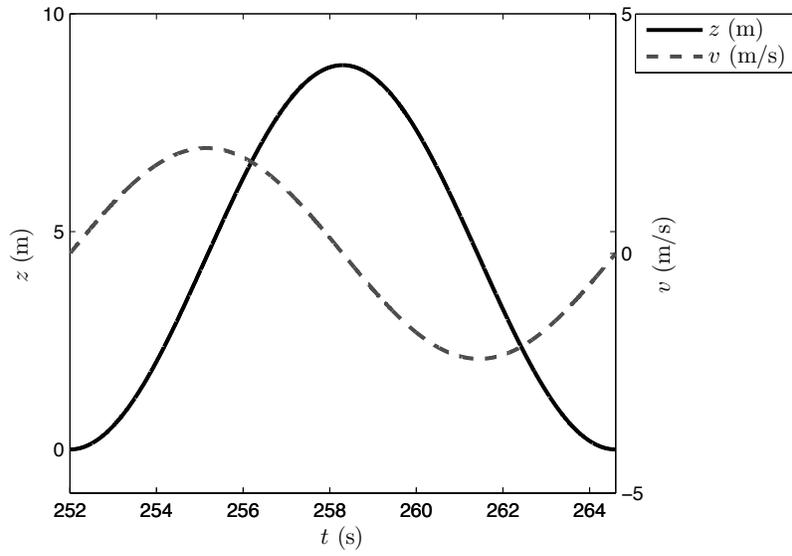


Figure 10: Simple Model Results for Position and Velocity vs. Time—One Period (see Fig. 9).

In Fig. 11 and Fig. 12, which depict current trajectories for all three phases, we can see that there is an rounded envelope shape that bounds the current trajectories. The three currents are symmetric about 0, with a maximum of 253.2 A and a minimum of -253.2 A. Figure 12 is scaled such that exactly one period of steady-state conditions is displayed.

As can be seen from the figures, the electrical dynamics are much faster than the mechanical dynamics (i.e., the position and velocity). The position and the velocity trajectories complete exactly one cycle during one period, whereas the current trajectories complete multiple cycles in one period. The maximum slope of the currents are measured to be approximately  $2.42 \times 10^4$  A/s, which represents a significant change rate. Also to be noted is that the plot presents a harmonic shape, with the currents being 0 at the beginning, middle and end of each period.

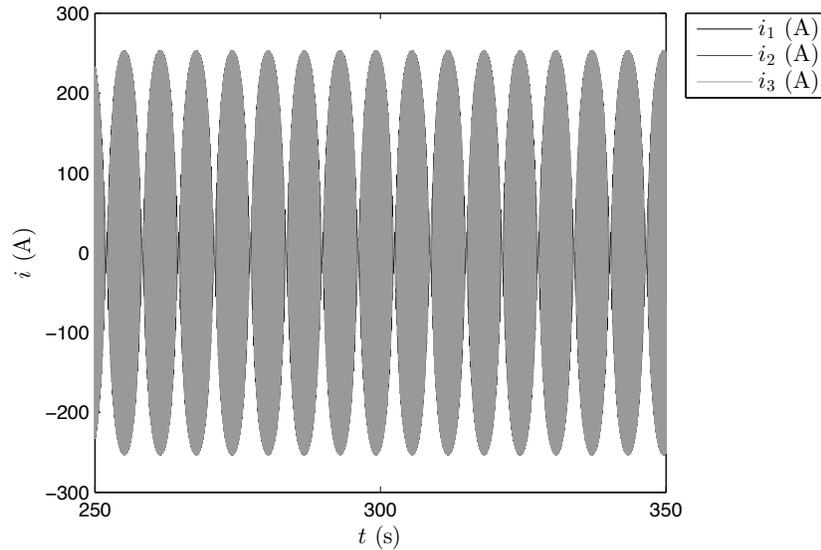


Figure 11: Simple Model Results for Current vs. Time.

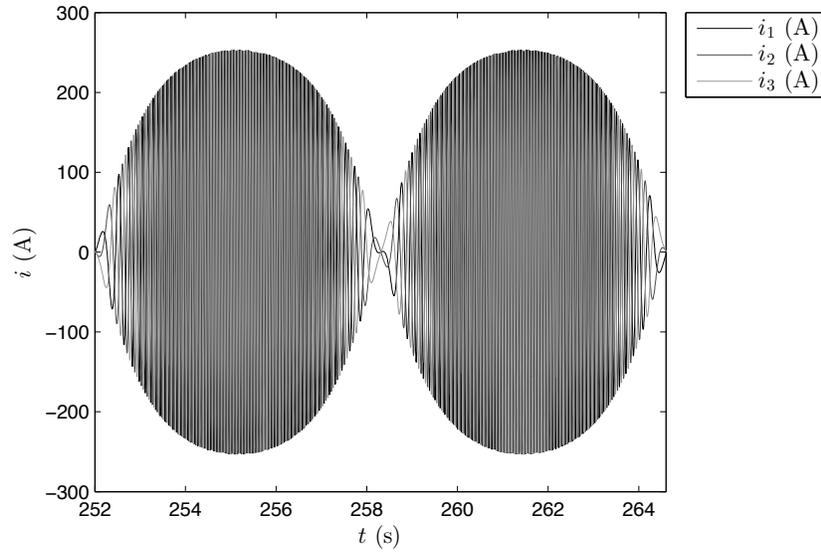


Figure 12: Simple Model Results for Current vs. Time—One Period (see Fig. 11).

Figures 13 and 14 describe the power output of the system, where Fig. 14 is scaled such that exactly one period of steady-state conditions is displayed. Power output ranges from 0 to  $6.1332 \times 10^5$  W. The power is non-negative since it is calculated using the formula  $P = i^2 R_L$ . During one period there are two peaks. The average power is  $3.7590 \times 10^5$  W.

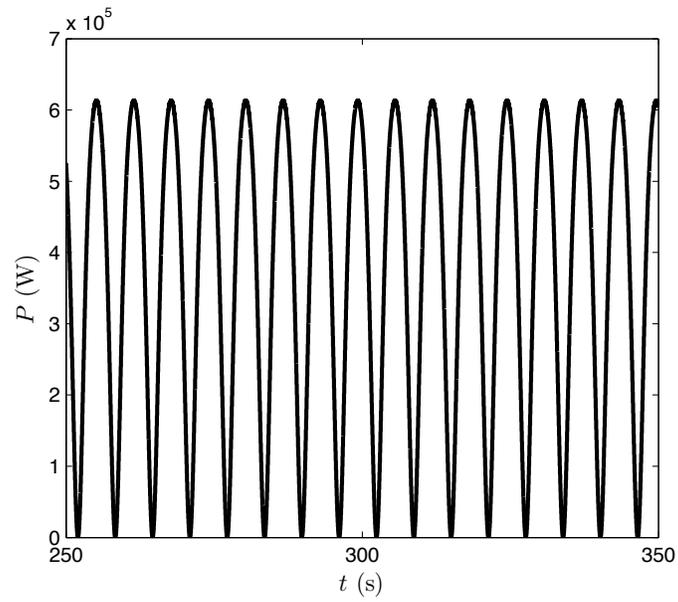


Figure 13: Simple Model Results for Power vs. Time.

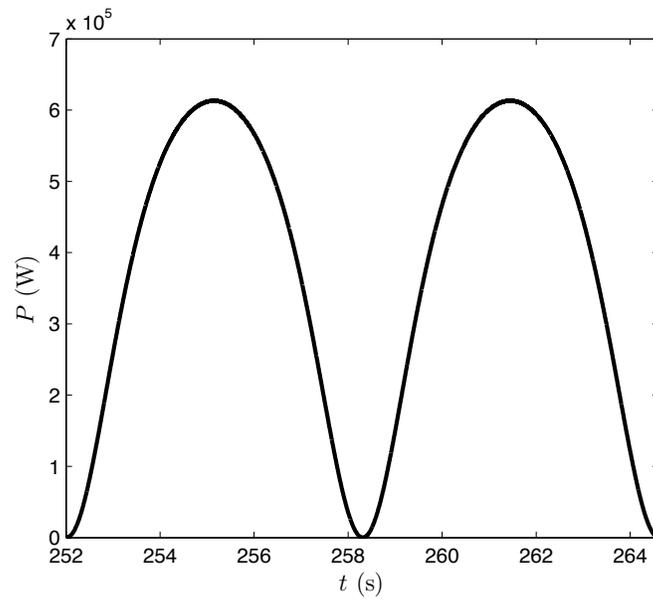


Figure 14: Simple Model Results for Power vs. Time—One Period (see Fig. 13).

### Complex Model Results

The complex model was then used in simulating the WEC. As described earlier, this model includes the electromagnetic force as quantified by Eqns. (2.21) and (2.22). The position  $z(t)$  and the velocity  $v(t)$ , depicted in Fig. 15 and Fig. 16, are still periodic. In addition, Fig. 16 shows that  $v$  is still close to harmonic, even with the added complexity of the system. The extremes of  $v(t)$  are  $-2.2236$  m/s and  $3.2256$  m/s. In terms of the position,  $z(t)$  is still close to harmonic, which is probably due to the damping term in the equations. The position trajectory  $z(t)$  has a range of  $-4.1688$ m to  $4.1688$ m.

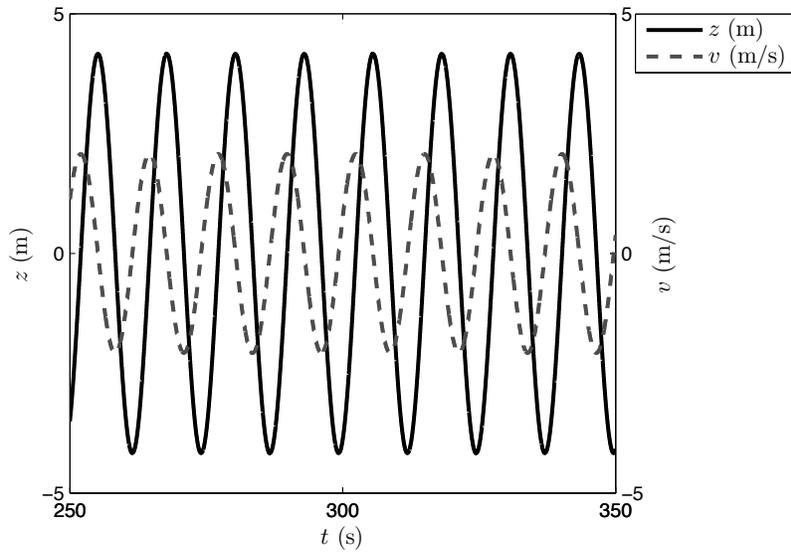


Figure 15: Complex Model Results for Position and Velocity vs. Time.

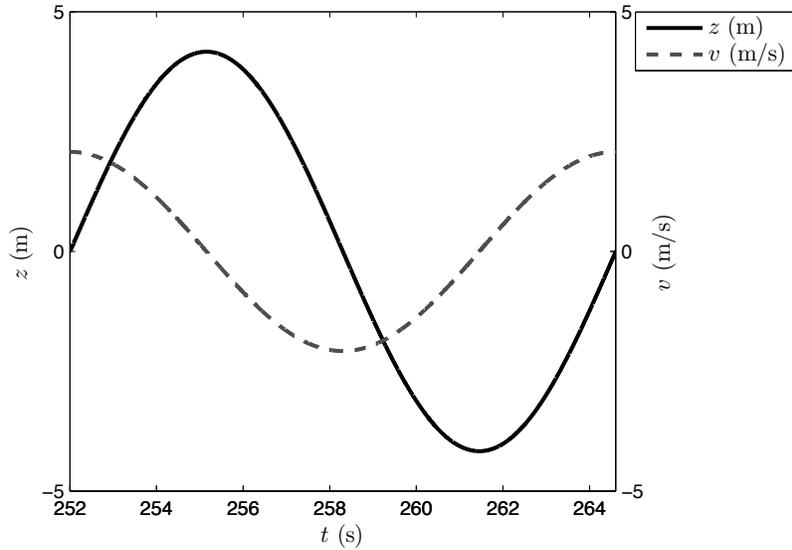


Figure 16: Complex Model Results for Position and Velocity vs. Time—One Period (see Fig. 15).

The current trajectories that resulted from the simulation of the complex model are shown in Figs. 17 and 18. The envelope of the plots in Fig. 17 is similar to the envelope for the simple model simulation results, but not identical because of the different dynamics of the buoy. The envelope still has the rounded shape as before. The range of  $i$  is -285 A to 285 A, which is also larger than the previous one. The dynamics are even faster with a maximum slope of  $4.00 \times 10^4$  A/s. The currents are still symmetric about 0.

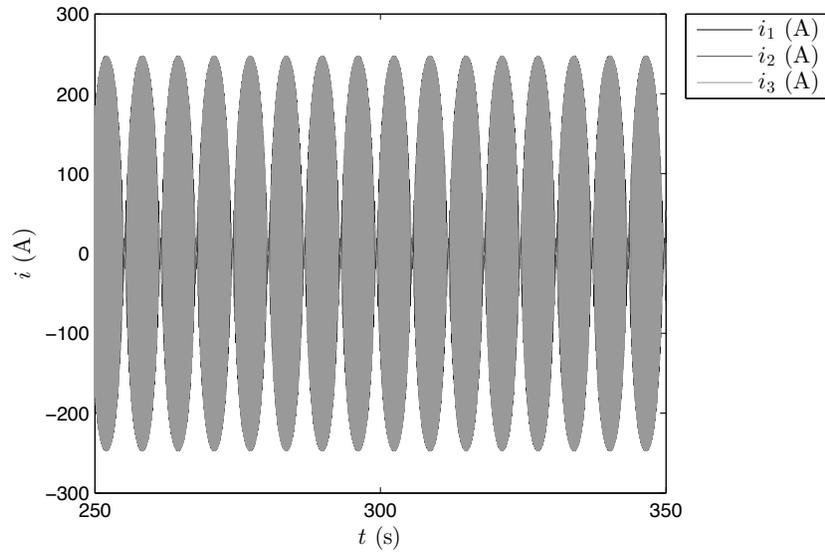


Figure 17: Complex Model Results for Current vs. Time.

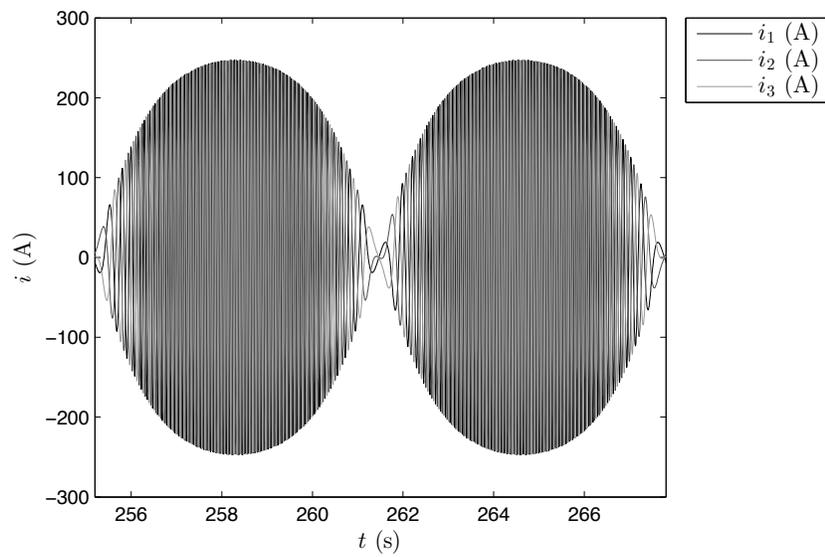


Figure 18: Complex Model Results for Current vs. Time—One Period (see Fig. 17).

The plots of  $P$ , namely Fig. 19 and Fig. 20, are also nearly the same as in the simple model simulation. The range, however, is 0 to  $7.7994 \times 10^5$  W with an average value of  $3.5319 \times 10^5$  W. While the maximum power output is larger than that of the simple model, the average power is slightly lower than that of the simple model. This indicates that while the power trajectories corresponding to the simple and complex model simulations appear to be the same upon visual inspection, they do in fact have different shapes. In other words, during a certain period of time, the area under the power output curve is smaller, indicating that the shape of the complex model power trajectory is more pointed.

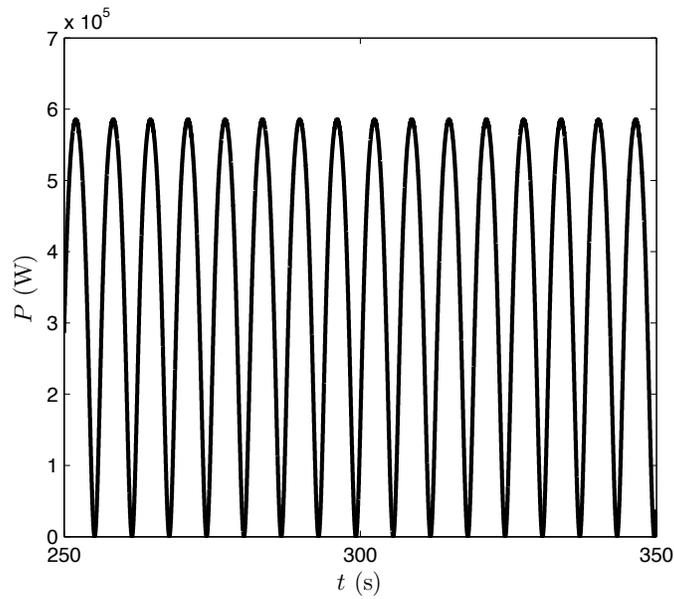


Figure 19: Complex Model Results for Power vs. Time.

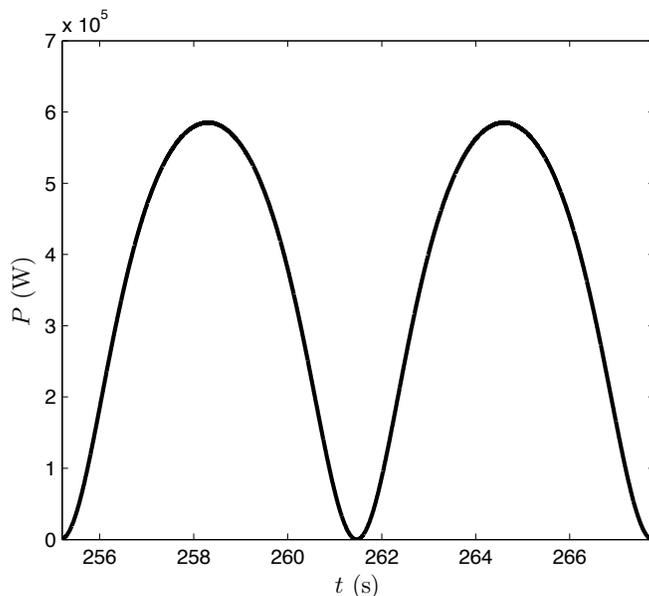


Figure 20: Complex Model Results for Power vs. Time—One Period (see Fig. 19).

### 3.2 Design Parameter Studies

A parametric study is an activity in which certain parameters are varied through a predetermined range, and the effect of changing these parameters is investigated. In the study presented here one parameter at a time is changed while other parameters are kept unchanged. For each parameter value combination to be tested a simulation is performed to determine the effect of certain parameters on the whole system.

In this report two parameters are studied using the complex model. The first parameter is the slot width  $b_s$ , and the second is the air gap  $g_a$ .

#### Slot Width Study

Various values of  $b_s$  are studied within the range of 0.022 m to 0.023 m. The maximum power is acquired around  $b_s = 0.02248$  m. Figure 21 illustrates energy extraction over one steady-state period as a function of  $b_s$ . It is depicted using a semi log plot as the energy values span several orders of magnitude. This plot exhibits a large peak with a maximum energy extraction value of  $9.0396 \times 10^{13}$  J.

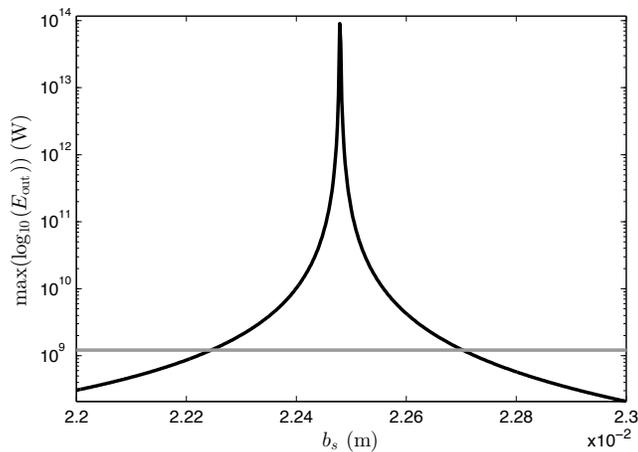


Figure 21: Slot Width ( $b_s$ ) Parameter Study (The horizontal line represents the maximum energy that can be extracted from the incoming ocean wave).

Figure 22 illustrates the power trajectory for the system design that corresponds to the maximum energy design in the slot width parametric study (i.e.,  $b_s = 0.02248$  m).

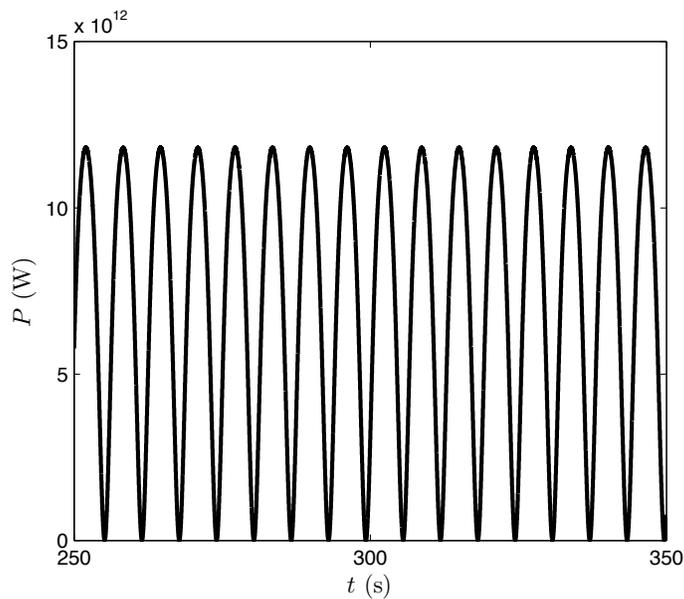


Figure 22: Power Trajectory for Slot Width Value Maximizing Peak Power of the Complex Model ( $b_s = 0.0225$  m).

The peak in the middle suggests resonant behavior. Given the magnitude of energy extraction at the peak, the validity of the model should be questioned. For example, for a given buoy size and wave conditions, it is possible to calculate the maximum amount of energy that can be extracted from an ocean wave. If all of the energy is extracted, the wave will be completely destroyed [11]. A practical wave energy converter will only be able to extract a fraction of the total available energy. Using the approach described by Herber and Allison [8], the maximum available energy in the wave was calculated and plotted as a horizontal line in Fig. 21.

Clearly values of  $b_s$  that produce energy values above this line in the simulation will not produce these energy values in a physical system. This model is not capturing some element of the real system behavior. Simply setting  $b_s = 0.02248$  m cannot be expected to produce optimal system behavior. In addition to improving model fidelity, system optimization would require the simultaneous consideration of all important design variables, not just a simple study of one parameter at a time<sup>2</sup>.

Figures 23 and 24 illustrate the current trajectories for the design where  $b_s$  is 0.0225 m. It can be observed that the current trajectories are very dynamic with larger ranges, meaning that more energy can be produced by this system design.

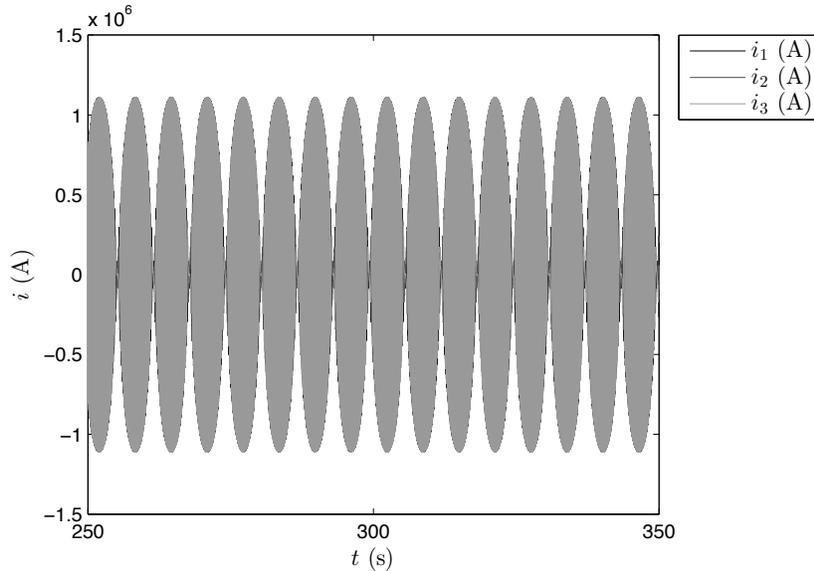


Figure 23: Current vs. Time for Slot Width Value Maximizing Peak Power of the Complex Model ( $b_s = 0.0225$  m).

<sup>2</sup>While such an optimization effort is beyond the scope of this report, it is the subject of ongoing research at the UIUC Engineering System Design Laboratory.

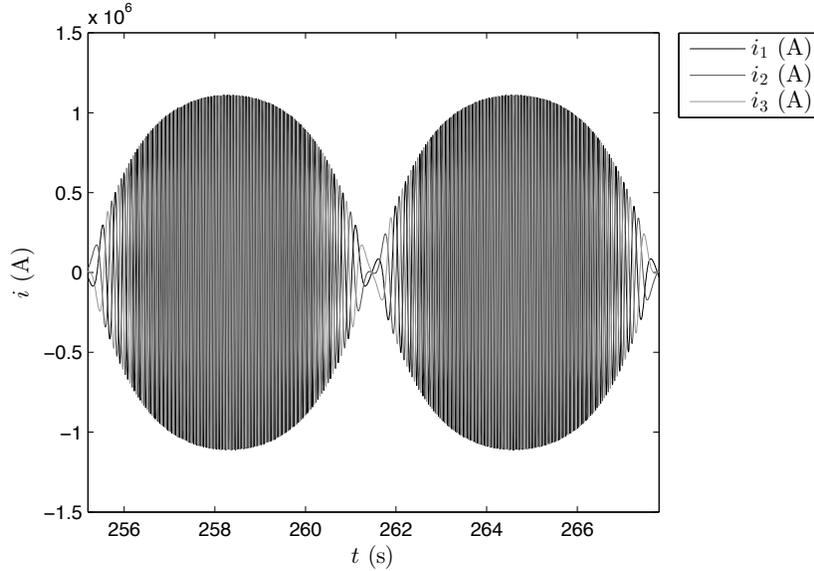


Figure 24: Current vs. Time for Slot Width Value Maximizing Peak Power of the Complex Model—One Period (see Fig. 23).

To summarize, according to the simulation results, the optimal slot width  $b_s$  is 0.0225 m. This will give a peak power of  $1.5741 \times 10^{13}$  W, according to Fig. 22. That produces  $9.0396 \times 10^{13}$  J per period. However, other constraints may prevent  $b_s$  from achieving the optimal value. The model here likely is not complete, and design variable interactions should be considered as well<sup>3</sup>.

### Air Gap Study

The second parameter studied here is the air gap  $g_a$ . Various values of  $g_a$  are studied within the range of 0.0035 m and 0.0045 m, and the maximum power is realized at  $g_a = 0.004$  m. The maximum energy extracted during a period is  $2.4494 \times 10^{13}$  J. The energy extraction per period as a function of  $g_a$  is illustrated in Fig. 25, and the power trajectory that corresponds to the maximum energy extraction is shown in Fig. 27.

<sup>3</sup>Determining the underlying cause of this model inaccuracy is beyond the scope of this report, and is a topic for future research.

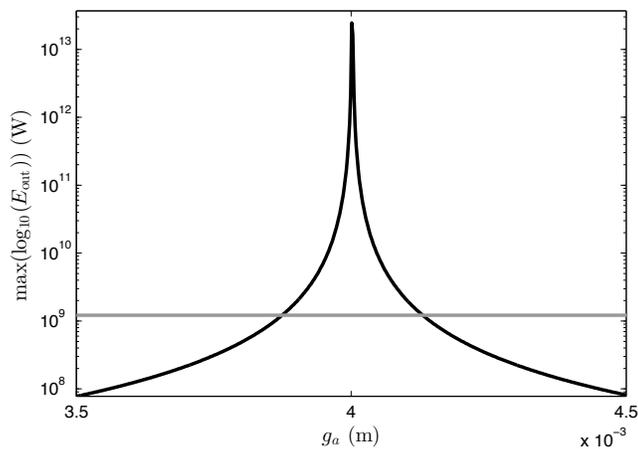


Figure 25: Air Gap ( $g_a$ ) Parameter Study (The horizontal line represents the maximum energy that can be extracted from the incoming ocean wave).

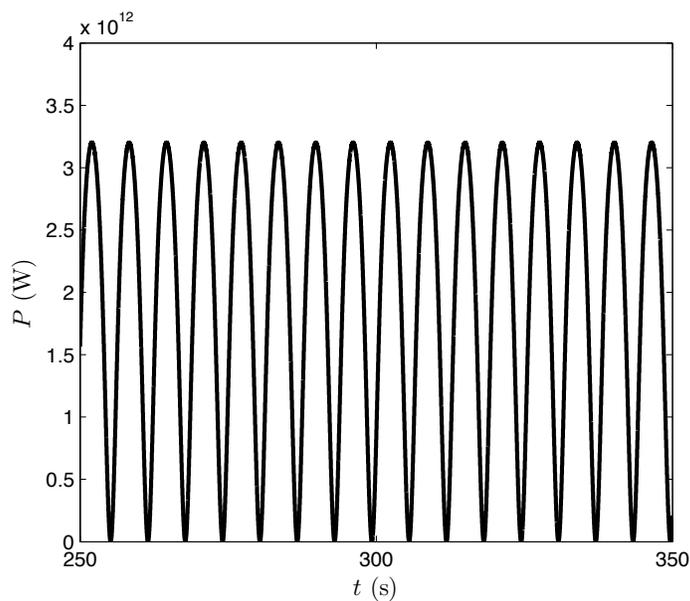


Figure 26: m

Figure 27: Power Trajectory for Air Gap Value Maximizing Peak Power of the Complex Model ( $g_a = 0.0040$  m).

As can be seen from Fig. 25, the range of  $g_a$  values that produce large energy

values is also very small. The peak in the middle also suggests that there is resonance occurring in the system when certain values of  $g_a$  are chosen. Again, the horizontal line represents the maximum amount of energy that can be possibly extracted from the incoming ocean wave, calculated according to Ref. [8]. This means that we actually can't reach the peak energy extraction value shown in this figure.

The current trajectories that correspond to the design that maximizes energy extraction (i.e.,  $g_a$  is 0.004 m) are illustrated in Figs. 28 and 29. The behavior here is very similar to what was observed in the slot width study. Ranges of resonance exist in both parametric studies, and the current trajectories that correspond to the peak energy extraction design have very large amplitudes. According to the model used here,  $g_a = 0.004$  m produces a maximum peak power of  $4.2683 \times 10^{12}$  W, and produces  $2.4494 \times 10^{13}$  J per period in steady state. As noted before, other constraints will prevent realization of the predicted maximum energy extraction.

Only two parameters of the generator are studied here. Other parameters, especially other independent physical variables, are a subject for further study.

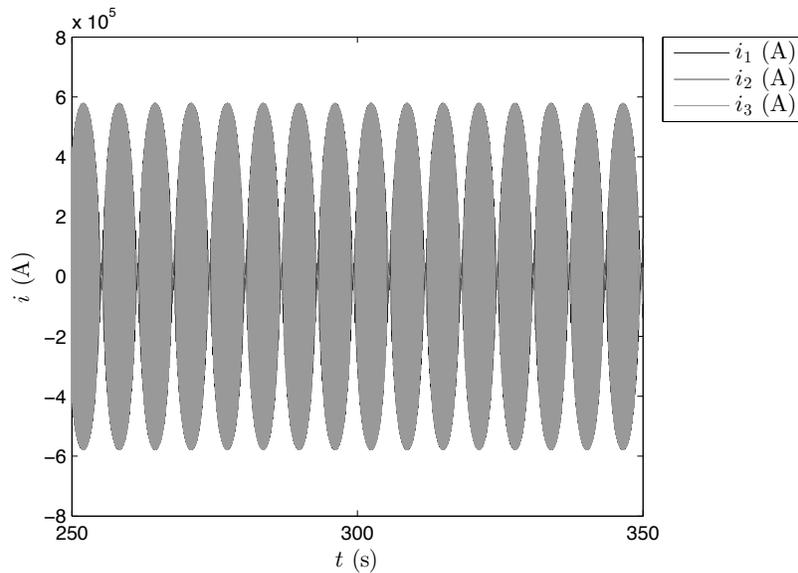


Figure 28: Current vs. Time for Air Gap Value Maximizing Peak Power of the Complex Model ( $g_a = 0.0040$  m).

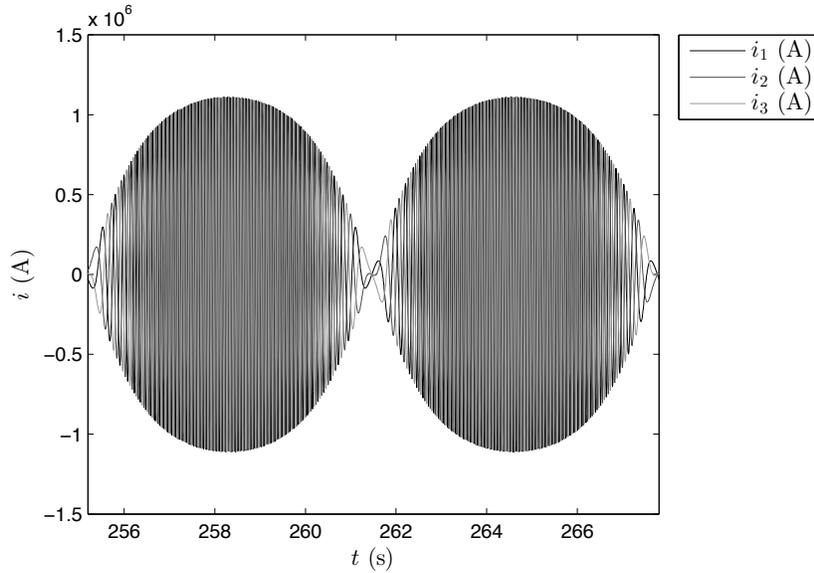


Figure 29: Current vs. Time for Air Gap Value Maximizing Peak Power of the Complex Model—One Period (see Fig. 28).

### 3.3 Model Stiffness Study

The stiffness of the system modeled in this report was also studied. A dynamic model is often thought of as stiff if it involves both fast and slow dynamics. Here the mechanical dynamics are much slower than the electrical dynamics. Another definition of model stiffness is that stiff systems may be simulated more efficiently using an implicit solver for ordinary differential equations (i.e., a stiff solver).

One way of determining stiffness is to evaluate the time steps required to simulate a system using both a stiff and non-stiff solver. Here `ode45` was the non-stiff solver used, and `ode15s` was the stiff solver used. Figure 30 illustrates how the time step sizes change through time for `ode45` based on the simple model, and Fig. 31 illustrates time step sizes for `ode15s` based on the simple model.

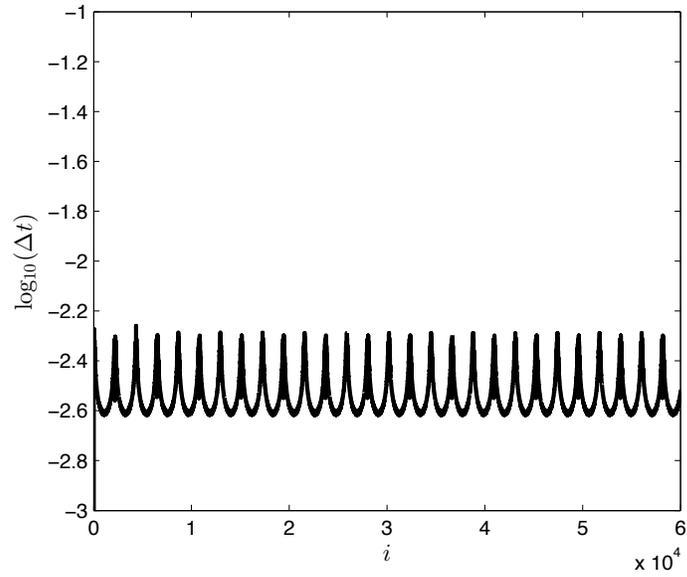


Figure 30: Plot of Time-Steps using the Simple Model with `ode45`.

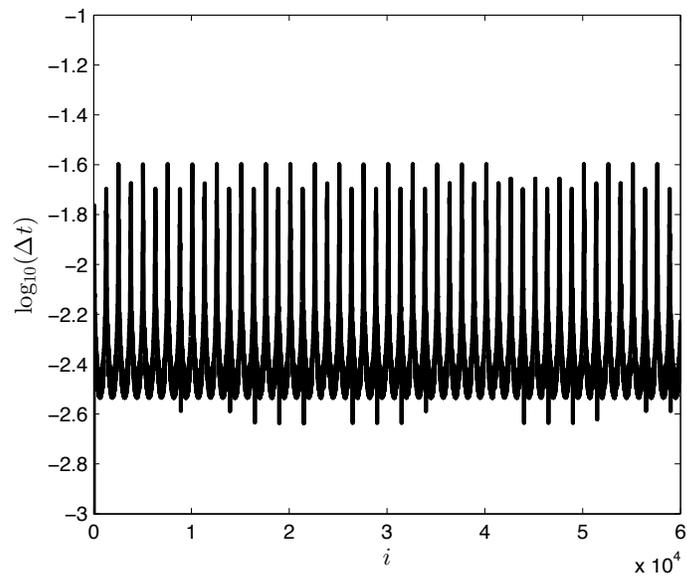


Figure 31: Plot of Time-Steps using the Simple Model with `ode15s`.

It is clear from Fig. 30 that the time steps remain very small (they differ by

less than an order of magnitude). This means that `ode45`, a non-stiff solver, but use very small time steps throughout the entire simulation to maintain the required accuracy. In other words, it struggles to solve the system of differential equations efficiently. Figure 31, on the other hand, shows that `ode15s` is able to use much larger time steps periodically and still maintain accuracy. After being driven to a small step size to maintain accuracy due to rapid change in state, the stiff solver is able to ‘recover’ to a larger step size, whereas the non-stiff solver is unable to maintain accuracy unless step sizes are kept very small. In other words, the stiff solver solves the system more efficiently than the non-stiff solver, so this is evidence that this system model exhibits stiffness. This is common in models of physical systems, and it is important to address the potential for dynamic stiffness when developing models for the design of physical systems.

A similar result was observed when comparing the `ode45` and `ode15s` solutions based on the complex model. These results are illustrated in Figs. 32 and 33. The range of step size values with the complex model is a bit larger than with the simple model, indicating somewhat increased system stiffness. In addition, the complex model takes some time to achieve steady-state time step behavior.

The total number of time steps for each simulation was also recorded. Table 3 summarizes these results. An additional non-stiff solver (`ode23`) was tested as well. Based on the number of required time steps, `ode23` performs similarly to `ode45`. For both the simple and complex models, the total number of required time steps (which is related to computational expense) is reduced significantly when switching to a stiff solver. This is additional evidence that the system model exhibits stiff behavior.

Table 3: Number of Time-Steps for Various ODE Solvers.

	<code>ode45</code>	<code>ode23</code>	<code>ode15s</code>
Simple	119708	116235	69655
Complex	117676	110316	66583

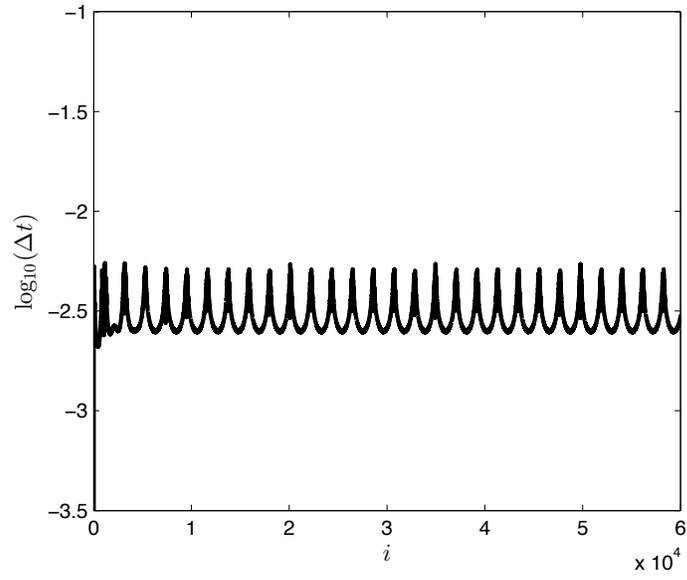


Figure 32: Plot of Time-Steps using the Complex Model with `ode45`.

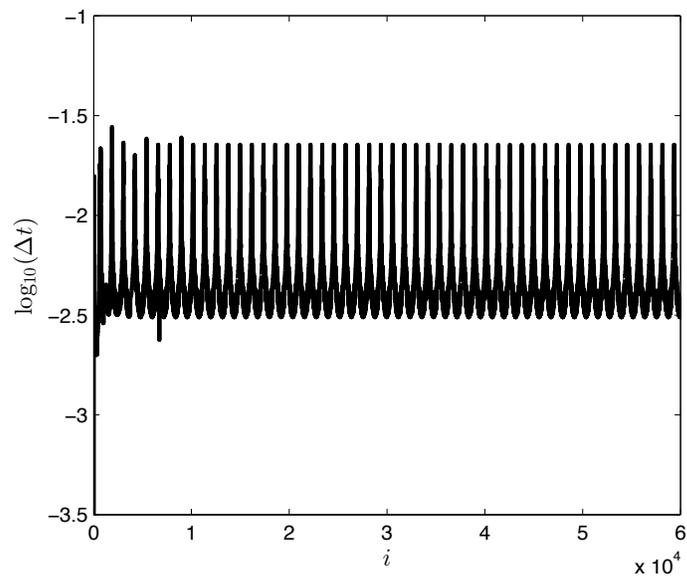


Figure 33: Plot of Time-Steps using the Complex Model with `ode15s`.

## 4 Conclusion

Wave energy extraction is of great potential importance given that fossil fuel as an energy source is limited and non-renewable. The design of a permanent magnet linear synchronous generator for wave energy extraction can greatly improve the efficiency. In this report the necessary physics theories for modeling and designing a permanent magnet linear synchronous generator are reviewed. Then a model of a permanent magnet linear synchronous generator is constructed, involving both independent and dependent parameters. It is simulated in MATLAB<sup>®</sup> for specific parameters. Two of the independent physical parameters of the generator were studied further using parametric sweeps. The results indicate that there is potential for improvements of the specific permanent magnet linear synchronous generator modeled in this work, as well as potential for improving the system model. Future work may focus on deeper study of the independent parameters of the model, interactions between independent parameters, improved-fidelity modeling (including more detailed models of other system components), and design optimization studies.

## Acknowledgments

This report is based on the prior work of James T. Allison, Allen Kaitharath and Daniel R. Herber [8, 9]. The work presented in this report was conducted under Prof. Allison’s supervision and with the guidance of Daniel R. Herber. Significant edits and feedback have been provided by both Prof. Allison and Daniel R. Herber.

## References

- [1] N.M. Kimoulakis, A.G. Kladas, and J.A. Tegopoulos. “Power Generation Optimization from Sea Waves by Using a Permanent Magnet Linear Generator Drive”. In: *IEEE Transactions on Magnetics* 44.6 (2008), pp. 1530–1533. DOI: 10.1109/TMAG.2007.914854.
- [2] R. Parthasarathy. “Linear PM Generator for Wave Energy Conversion”. Master’s Thesis. Anna University, 2012. URL: <http://etd.lsu.edu/docs/available/etd-04182012-121521/unrestricted/parthasarathythesis.pdf>.
- [3] S.A. Nasar and I. Boldea. *Linear Electric Motors: Theory, Design and Practical Applications*. Prentice Hall, 1987. ISBN: 9780135368633.
- [4] O. Danielsson. “Design of a Linear Generator for Wave Energy Plant”. Master’s Thesis. Uppsala University, 2003. URL: <http://www.el.angstrom.uu.se/meny/artiklar/design%20of%20a%20linear%20generator%20for%20wave%20energy%20plant4.pdf>.
- [5] A.W. van Zyl et al. “Comparison of Force to Weight Ratios Between a Single-Sided Linear Synchronous Motor and a Tubular Linear Synchronous Motor”. In: *IEEE International Electric Machines and Drives Conference 1999*. Seattle, WA, USA, 1999, pp. 571–573. ISBN: 0780352939. DOI: 10.1109/IEMDC.1999.769178.
- [6] F.W. Carter. “Note on Air-Gap and Interpolar Induction”. In: *Journal of the Institution of Electrical Engineers* 29 (1900), pp. 925–933. DOI: 10.1049/jiee-1.1900.0095.
- [7] F.W. Carter. “Air-Gap Induction”. In: *Electric World and Engineer* 38.22 (1901), pp. 884–888. URL: <http://hdl.handle.net/2027/uva.x030741299>.
- [8] J.T. Allison and D.R. Herber. “Wave Energy Extraction Maximization in Irregular Ocean Waves Using Pseudospectral Methods”. In: *ASME 2013 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*. Portland, OR, USA, 2013. URL: <http://systemdesign.illinois.edu/publications/Her13a.pdf>.

- [9] J.T. Allison, A. Kaitharath, and D.R. Herber. “Wave Energy Extraction Maximization Using Direct Transcription”. In: *ASME 2012 International Mechanical Engineering Congress and Exposition*. Houston, TX, USA, 2012. URL: <http://systemdesign.illinois.edu/publications/A1112c.pdf>.
- [10] T. Jeans et al. “Irregular Deep Ocean Wave Energy Conversion Using a Cycloidal Wave Energy Converter”. In: *9th European Wave and Tidal Energy Conference*. Southampton, UK, 2011.
- [11] Johannes Falnes. *Wave-Energy Absorption by Oscillating Bodies*. Cambridge University Press, 2002, pp. 196–224.

## A Appendix: Modeling Code

Simulation.m

```
1 % This script accepts the values for the independent variables ...
   bs and ga
2 % and also a flag indicating which model should be used. x(1) is the
3 % position of the buoy. x(2) is the velocity of the buoy. x(3), ...
   x(4) and
4 % x(5) are the three currents. The outputs are t, x and Pout, ...
   which is the
5 % output power.
6 function [t,x,Pout] = Simulation(bs,ga,flag)
7
8     % run simulation
9     [t,x] = ode45(@ (t,x) LPM_3Phase(t,x,bs,ga,flag), [0 350], [0 ...
   0 0 0 0]);
10
11     % calculate power
12     RL = 7.5; %Ohm
13     v(:,1) = x(:,3)*RL;
14     v(:,2) = x(:,4)*RL;
15     v(:,3) = x(:,5)*RL;
16     Pin = x(:,3).*v(:,1) + x(:,4).*v(:,2) + x(:,5).*v(:,3);
17     Pout = Pin.*0.85;
18
19 end
20
21 function dx = LPM_3Phase(t,x,bs,ga,flag)
22     dx = zeros(5,1);
23     p = 6;
24     q = 1;
25     m = 3;
26     N = zeros(1,3);
27     for i = 1:3
28         N(i) = 82*p*q;
29     end
30     L = 0.432; %m
31     taup = L/p; %m
32     taut = taup./(m*q); %m
33     % bs = 0.016; %m
34     bt = taut - bs; %m
35     % ga = 0.002; %m
36     Kc = taut*(5*ga+bs)/(taut*(5*ga+bs)-bs^2);
37     geq = Kc*ga;
38     hm = 0.006; %m
39     Br = 1.2; %T
40     mu0 = 4*pi*10^(-7);
41     Hc = 905000;
42     phi = zeros(1,3);
43     for i = 1:3
44         phi(i) = (Br*hm*Hc*mu0)/(hm*Hc*mu0 - geq*Br);
45     end
46
47     if flag == 1 % Simple
48         um = 2.2; %m/s
```

```

49     uav = 2/pi*um; %m/s
50     T = 12.6; %s
51     Ra = 1.5; %Ohm
52     Ls = 0.115; %H
53     RL = 7.5; %Ohm
54     Ms = 4; %number of armatures
55     Ws = 0.2; %m
56     KE = Ms*Ws*N(i)*phi(i)*uav;
57     % u = um*sin(omegam*t);
58     u = um.*sin(2*pi.*t./T);
59     du = 2*pi.*um./T*cos(2*pi.*t./T);
60     e = zeros(1,3);
61     e(1) = KE*cos(pi.*x(1)./taup)*u;
62     e(2) = KE*cos(pi.*x(1)./taup - 2*pi/3)*u;
63     e(3) = KE*cos(pi.*x(1)./taup - 4*pi/3)*u;
64     v = zeros(1,3);
65     v(1) = x(3)*RL;
66     v(2) = x(4)*RL;
67     v(3) = x(5)*RL;
68     F = zeros(1,3);
69     for i = 1:3
70         F(i) = ...
            N(i).*phi(i).*pi./taup.*sin(pi.*x(1)./taup).*x(i ...
            + 2);
71     end
72     dx(1) = u; % u
73     dx(2) = du;
74     elseif flag == 2 % Complex
75         um = 2.2; %m/s
76         uav = 2/pi*um; %m/s
77         T = 12.6; %s
78         A = 4; %um*T/(2*pi); %m
79         Ra = 1.5; %Ohm
80         Ls = 0.115; %H
81         RL = 7.5; %Ohm
82         Ms = 4; %number of armatures
83         Ws = 0.2; %m
84         KE = Ms*Ws*N(i)*phi(i)*uav;
85         % u = um*sin(omegam*t);
86         u = x(2);
87         e = zeros(1,3);
88         e(1) = KE*cos(pi.*x(1)./taup).*u;
89         e(2) = KE*cos(pi.*x(1)./taup - 2*pi/3).*u;
90         e(3) = KE*cos(pi.*x(1)./taup - 4*pi/3).*u;
91         v = zeros(1,3);
92         v(1) = x(3)*RL;
93         v(2) = x(4)*RL;
94         v(3) = x(5)*RL;
95         F = zeros(1,3);
96         for i = 1:3
97             F(i) = ...
                N(i).*phi(i).*pi./taup.*sin(pi.*x(1)./taup).*x(i ...
                + 2);
98         end
99         rho = 1025; %kg/m^3
100        g = 9.81; %m/s^2
101        a = 0.5; %m

```

```

102     b = 1; %m
103     h = 10; %m
104     M = 1000; %kg
105     mr = 289; %kg
106     Rr = 16; %Ns/m
107     Rv = 717; %Ns/m
108     dx(1) = x(2);
109     dx(2) = -((Rr + Rv).*(x(2) - ...
110             2.*pi.*A./T.*cos(2.*pi.*t./T)) + ...
111             rho.*g.*pi.*a.^2.*(x(1) - A.*sin(2.*pi.*t./T)))./(M ...
112             + mr);
111     end
112     dx(3) = (e(1) - v(1) - x(3)*Ra)/Ls;
113     dx(4) = (e(2) - v(2) - x(4)*Ra)/Ls;
114     dx(5) = (e(3) - v(3) - x(5)*Ra)/Ls;
115 end

```

#### Parameter \_ Studies.m

```

1 % This script will run the simulation for 500 different values ...
2 % within the
3 % range given for two independent variables, bs and ga. The ...
4 % results, which
5 % is the power output, will be analyzed and plotted.
6 path = msavename(mfilename('fullpath'),'SavedData');
7 fig_index = 0;
8
9 %% bs study
10 start = 0.022;
11 finish = 0.023;
12 N = 500;
13 array = linspace(start,finish,N);
14
15 results = zeros(N,1);
16 parfor i = 1:N
17     bs = array(i); % m
18     ga = 0.002; % m
19     [t,x,Pout] = Simulation(bs,ga,2);
20
21     t1 = find(t <= 250);
22     i1 = find(t1 == t1(end));
23     T = 12.6; %s
24     n = (t(end) - t(i1))./T;
25     periodT = t(i1:end);
26     periodP = Pout(i1:end);
27     E = trapz(periodT, periodP)./n;
28     results(i) = E; % max(Pout);
29 end
30 save(strcat(path, 'bs-study'))
31
32 %% figure // bs-study
33 %load(strcat(path, 'bs-study'))
34 data.fig_name = 'bs-study';
35 % label names

```

```

36 data.xname = '$b_s$ (m)';
37 data.yname = '$\max(P_{\mathrm{out}})$ (W)';
38 % state data
39 data.x1 = array;
40 data.y1 = results; % position
41 % axis limits
42 data.xmin = start;
43 data.xmax = finish;
44 data.ymin = min(results);
45 data.ymax = max(results);
46
47 fig_index = fig_index + 1;
48 create_figure(data, fig_index, 1)
49 save2pdf(strcat(path, data.fig_name, '.pdf'), fig_index, 600)
50
51 %% figure // Power-Complex-bs
52 % Simulate using optimal bs
53 bs_index = find(results == max(results));
54 bs = bs_index; % m
55 ga = 0.002; % m
56 [t, x, Pout] = Simulation(bs, ga, 2);
57
58 data.fig_name = 'Power-Complex-bs';
59 % label names
60 data.xname = '$t$ (s)';
61 data.yname = '$P$ (W)';
62 % state data
63 data.x1 = t;
64 data.y1 = Pout; % power
65 % axis limits
66 data.xmin = 250;
67 data.xmax = 350;
68 data.ymin = 0;
69 data.ymax = 1.5*10^13;
70
71 fig_index = fig_index + 1;
72 create_figure(data, fig_index, 1)
73 save2pdf(strcat(path, data.fig_name, '.pdf'), fig_index, 600)
74
75 %% figure // Buoy-Complex-bs
76 % Simulate using optimal bs
77 bs_index = find(results == max(results));
78 bs = bs_index; % m
79 ga = 0.002; % m
80 [t, x, Pout] = Simulation(bs, ga, 2);
81
82 data.fig_name = 'Buoy-Complex-bs';
83 % label names
84 data.xname = '$t$ (s)';
85 data.y1name = '$z$ (m)';
86 data.y2name = '$v$ (m/s)';
87 % state data
88 data.x1 = t;
89 data.x2 = t;
90 data.y1 = x(:, 1); % position
91 data.y2 = x(:, 2); % speed
92 % axis limits

```

```

93 data.xmin = 250;
94 data.xmax = 350;
95 data.ylmin = -5;
96 data.ylmax = 5;
97 data.y2min = -5;
98 data.y2max = 5;
99
100 fig_index = fig_index + 1;
101 create_figure(data, fig_index, 2)
102 save2pdf(strcat(path, data.fig_name, '.pdf'), fig_index, 600)
103
104 %% figure // Buoy-Complex-Zoom-bs
105 % Simulate using optimal bs
106 bs_index = find(results == max(results));
107 bs = bs_index; % m
108 ga = 0.002; % m
109 [t, x, Pout] = Simulation(bs, ga, 2);
110
111 data.fig_name = 'Buoy-Complex-Zoom-bs';
112 % label names
113 data.xname = '$t$ (s)';
114 data.y1name = '$z$ (m)';
115 data.y2name = '$v$ (m/s)';
116 % state data
117 data.x1 = t;
118 data.x2 = t;
119 data.y1 = x(:, 1); % position
120 data.y2 = x(:, 2); % speed
121 % axis limits
122 data.xmin = 252;
123 data.xmax = 264.6;
124 data.ylmin = -5;
125 data.ylmax = 5;
126 data.y2min = -5;
127 data.y2max = 5;
128
129 fig_index = fig_index + 1;
130 create_figure(data, fig_index, 2)
131 save2pdf(strcat(path, data.fig_name, '.pdf'), fig_index, 600)
132
133 %% figure // Current-Complex-bs
134 % Simulate using optimal bs
135 bs_index = find(results == max(results));
136 bs = bs_index; % m
137 ga = 0.002; % m
138 [t, x, Pout] = Simulation(bs, ga, 2);
139
140 data.fig_name = 'Current-Complex-bs';
141 % label names
142 data.xname = '$t$ (s)';
143 data.yname = '$i$ (A)';
144 data.y1name = '$i.1$ (A)';
145 data.y2name = '$i.2$ (A)';
146 data.y3name = '$i.3$ (A)';
147 % state data
148 data.x1 = t;
149 data.y1 = x(:, 3); % current 1

```

```

150 data.y2 = x(:,4); % current 2
151 data.y3 = x(:,5); % current 3
152 % axis limits
153 data.xmin = 250;
154 data.xmax = 350;
155 data.ymin = -1.5*10^6;
156 data.ymax = 1.5*10^6;
157
158 fig_index = fig_index + 1;
159 create_figure(data,fig_index,3)
160 save2pdf(strcat(path,data.fig_name,'.pdf'),fig_index,600)
161
162 %% figure // Current-Complex-Zoom-bs
163 % Simulate using optimal bs
164 bs_index = find(results == max(results));
165 bs = bs_index; % m
166 ga = 0.002; % m
167 [t,x,Pout] = Simulation.Complex(bs,ga,2);
168
169 data.fig_name = 'Current-Complex-Zoom-bs';
170 % label names
171 data.xname = '$t$ (s)';
172 data.yname = '$i$ (A)';
173 data.y1name = '$i_1$ (A)';
174 data.y2name = '$i_2$ (A)';
175 data.y3name = '$i_3$ (A)';
176 % state data
177 data.x1 = t;
178 data.y1 = x(:,3); % current 1
179 data.y2 = x(:,4); % current 2
180 data.y3 = x(:,5); % current 3
181 % axis limits
182 data.xmin = 255.2;
183 data.xmax = 267.8;
184 data.ymin = -1.5*10^6;
185 data.ymax = 1.5*10^6;
186
187 fig_index = fig_index + 1;
188 create_figure(data,fig_index,3)
189 save2pdf(strcat(path,data.fig_name,'.pdf'),fig_index,600)
190
191 %% ga study
192 start = 0.0035;
193 finish = 0.0045;
194 N = 500;
195 array = linspace(start,finish,N);
196
197 results = zeros(N,1);
198 parfor i = 1:N
199
200     bs = 0.016; % m
201     ga = array(i); % m
202     [t,x,Pout] = Simulation(bs,ga,2);
203
204     t1 = find(t ≤ 250);
205     i1 = find(t1 == t1(end));
206     T = 12.6; %s

```

```

207     n = (t(end) - t(i1))./T;
208     periodT = t(i1:end);
209     periodP = Pout(i1:end);
210     E = trapz(periodT, periodP)./n;
211     results(i) = E; % max(Pout);
212
213 end
214 save(strcat(path, 'ga-study'))
215
216 %% figure // ga-study
217 %load(strcat(path, 'bs-study'))
218 data.fig_name = 'ga-study';
219 % label names
220 data.xname = '$g_a$ (m)';
221 data.yname = '$\max(P_{\mathrm{out}})$ (W)';
222 % state data
223 data.x1 = array;
224 data.y1 = results; % position
225 % axis limits
226 data.xmin = start;
227 data.xmax = finish;
228 data.ymin = min(results);
229 data.ymax = 8.6*10^12;
230
231 fig_index = fig_index + 1;
232 create_figure(data, fig_index, 1)
233 save2pdf(strcat(path, data.fig_name, '.pdf'), fig_index, 600)
234
235 %% figure // Power-Complex-ga
236 % Simulate using optimal ga
237 ga_index = find(results == max(results));
238 bs = 0.016; % m
239 ga = ga_index; % m
240 [t,x,Pout] = Simulation(bs, ga, 2);
241
242 data.fig_name = 'Power-Complex-ga';
243 % label names
244 data.xname = '$t$ (s)';
245 data.yname = '$P$ (W)';
246 % state data
247 data.x1 = t;
248 data.y1 = Pout; % power
249 % axis limits
250 data.xmin = 250;
251 data.xmax = 350;
252 data.ymin = 0;
253 data.ymax = 4*10^12;
254
255 fig_index = fig_index + 1;
256 create_figure(data, fig_index, 1)
257 save2pdf(strcat(path, data.fig_name, '.pdf'), fig_index, 600)
258
259 %% figure // Buoy-Complex-ga
260 % Simulate using optimal ga
261 ga_index = find(results == max(results));
262 bs = 0.016; % m
263 ga = ga_index; % m

```

```

264 [t,x,Pout] = Simulation(bs,ga,2);
265
266 data.fig_name = 'Buoy-Complex-ga';
267 % label names
268 data.xname = '$t$ (s)';
269 data.y1name = '$z$ (m)';
270 data.y2name = '$v$ (m/s)';
271 % state data
272 data.x1 = t;
273 data.x2 = t;
274 data.y1 = x(:,1); % position
275 data.y2 = x(:,2); % speed
276 % axis limits
277 data.xmin = 250;
278 data.xmax = 350;
279 data.y1min = -5;
280 data.y1max = 5;
281 data.y2min = -5;
282 data.y2max = 5;
283
284 fig_index = fig_index + 1;
285 create_figure(data,fig_index,2)
286 save2pdf(strcat(path,data.fig_name,'.pdf'),fig_index,600)
287
288 %% figure // Buoy-Complex-Zoom-ga
289 % Simulate using optimal ga
290 ga_index = find(results == max(results));
291 bs = 0.016; % m
292 ga = ga_index; % m
293 [t,x,Pout] = Simulation(bs,ga,2);
294
295 data.fig_name = 'Buoy-Complex-Zoom-ga';
296 % label names
297 data.xname = '$t$ (s)';
298 data.y1name = '$z$ (m)';
299 data.y2name = '$v$ (m/s)';
300 % state data
301 data.x1 = t;
302 data.x2 = t;
303 data.y1 = x(:,1); % position
304 data.y2 = x(:,2); % speed
305 % axis limits
306 data.xmin = 252;
307 data.xmax = 264.6;
308 data.y1min = -5;
309 data.y1max = 5;
310 data.y2min = -5;
311 data.y2max = 5;
312
313 fig_index = fig_index + 1;
314 create_figure(data,fig_index,2)
315 save2pdf(strcat(path,data.fig_name,'.pdf'),fig_index,600)
316
317 %% figure // Current-Complex-ga
318 % Simulate using optimal ga
319 ga_index = find(results == max(results));
320 bs = 0.016; % m

```

```

321 ga = ga_index; % m
322 [t,x,Pout] = Simulation(bs,ga,2);
323
324 data.fig_name = 'Current-Complex-ga';
325 % label names
326 data.xname = '$t$ (s)';
327 data.yname = '$i$ (A)';
328 data.y1name = '$i.1$ (A)';
329 data.y2name = '$i.2$ (A)';
330 data.y3name = '$i.3$ (A)';
331 % state data
332 data.x1 = t;
333 data.y1 = x(:,3); % current 1
334 data.y2 = x(:,4); % current 2
335 data.y3 = x(:,5); % current 3
336 % axis limits
337 data.xmin = 250;
338 data.xmax = 350;
339 data.ymin = -0.8*10^6;
340 data.ymax = 0.8*10^6;
341
342 fig_index = fig_index + 1;
343 create_figure(data,fig_index,3)
344 save2pdf(strcat(path,data.fig_name,'.pdf'),fig_index,600)
345
346 %% figure // Current-Complex-Zoom-ga
347 % Simulate using optimal ga
348 ga_index = find(results == max(results));
349 bs = 0.016; % m
350 ga = ga_index; % m
351 [t,x,Pout] = Simulation(bs,ga,2);
352
353 data.fig_name = 'Current-Complex-Zoom-ga';
354 % label names
355 data.xname = '$t$ (s)';
356 data.yname = '$i$ (A)';
357 data.y1name = '$i.1$ (A)';
358 data.y2name = '$i.2$ (A)';
359 data.y3name = '$i.3$ (A)';
360 % state data
361 data.x1 = t;
362 data.y1 = x(:,3); % current 1
363 data.y2 = x(:,4); % current 2
364 data.y3 = x(:,5); % current 3
365 % axis limits
366 data.xmin = 255.2;
367 data.xmax = 267.8;
368 data.ymin = -0.8*10^6;
369 data.ymax = 0.8*10^6;
370
371 fig_index = fig_index + 1;
372 create_figure(data,fig_index,3)
373 save2pdf(strcat(path,data.fig_name,'.pdf'),fig_index,600)

```