Control Design and Model Validation Applications in Nonlinear Vessel Dynamics

Michele D Cooper

Dissertation submitted to the Faculty of
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy
in
Aerospace Engineering

Leigh S. McCue-Weil, Chair
Alan J. Brown
Shane D. Ross
Craig A. Woolsey

May 4, 2015
Blacksburg, VA

Keywords: Nonlinear Dynamics, Ship, Chaos, Model Validation, Non-intrusive Polynomial Chaos, Control

ABSTRACT

Control Design and Model Validation for Topics in Nonlinear Vessel Dynamics

Michele D Cooper

Chair: Leigh McCue

In recent decades, computational models have become critical to how engineers and mathematicians understand nature; as a result they have become an integral part of the design process in most engineering disciplines. Moore's law anticipates computing power doubling every two years; a prediction that has historically been realized. As modern computing power increases, problems that were previously too complex to solve by hand or by previous computing abilities become tractable. This has resulted in the development of increasingly complex computational models simulating increasingly complex dynamics. Unfortunately, this has also resulted in increased challenges in fields related to model development, such as model validation and model based control, which are needed to make models useful in the real world.

Much of the validation literature to date has focused on spatial and spatiotemporal simulations; validation approaches are well defined for such models. For most time series simulations, simulated and experimental trajectories can be directly compared negating the need for specialized validation tools. In the study of some ship motion behavior, chaos exists, which results in chaotic time series simulations. This presents novel challenges for validation; direct comparison may not be the most apt approach.

For these applications, there is a need to develop appropriate metrics for model validation. A major thrust of the current work seeks to develop a set of validation metrics for such chaotic time series data. A complementary but separate portion of work investigates Non-Intrusive Polynomial Chaos as an approach to reduce the computational costs associated with uncertainty analysis and other stochastic investigations into the behavior of nonlinear, chaotic models.

A final major thrust of this work focuses on contributing to the control of nonlinear marine systems, specifically the autonomous recovery of an unmanned surface vehicle utilizing motion prediction information. The same complexity and chaotic nature that makes the validation of ship motion models difficult can also make the development of reliable, robust controllers difficult as well. This body of work seeks to address several facets of this broad need that has developed due to our increased computational abilities by providing validation metrics and robust control laws.

To my husband, Daniel, who has supported me throughout my academic wanderings.

ACKNOWLEDGEMENTS

I would like to thank my advisor Dr. Leigh McCue, who accepted an academic wanderer into her group and introduced me to the world of ship dynamics. The journey has been long and she has been endlessly patient with me. Along with a few forceful nudges, she has helped me better myself as a researcher and as a person. I would also like to thank my committee members Drs. Alan Brown, Shane Ross and Craig Woolsey. They have proven helpful and supportive throughout this endeavor and I have appreciated all the helpful questions and guidance as well as understanding they have provided.

I would also like to express my gratefulness to the sponsors who supported this work from the programs of Drs. Pat Purtell, Ki-Han Kim, and Thomas Fu at the Office of Naval Research under grant number N00014-10-1-0398, Dr. Robert Brizzolara of the Office of Naval Research under grant number N00014-10-1-0582, and Dr. Eduardo Misawa at the National Science Foundation under grant CMMI 0747973; without their generous support none of this research could have been conducted.

I would also like to thank Mr. Kenneth Weems for his knowledge and expertise as well as Dr. Pavlos Vlachos for allowing us use of his facilities while conducting the wedge entry experiment.

TABLE OF CONTENTS

LIST OF FIGURES

Figure

LIST OF ABBREVIATIONS

CHAPTER I

Introduction

In today's technology driven age, the world has access to powerful machines that allow scientists to tackle larger and more complex problems than ever before. However, before the development of such computers, numerical modelling was limited to very simple systems, often solved by hand or at best by mechanical computers. Algorithms for dealing with nonlinear phenomena numerically existed for many years, but the scale of the problems that could be investigated has been extremely limited. For example, in 1940 Atanasoff described the use of a machine which would speed up the solution of a set of 20 coupled linear equations so that a solution might be obtained in twelve hours, a drastic improvement over the 125 hours he anticipated the hand calculation to require (Atanasoff, 1940). Under these conditions, where a simple, linear system took twelve hours (with state of the art technology at the time) to solve, solving infinitely more complex systems with nonlinear and chaotic behavior was, quite simply, impossible. The investigation of such systems resided solely in the realm of the experimentalist. However, a dramatic shift was coming.

The pressures of World War II lead to the development of the world'first electronic computer – ENIAC. ENIAC found rapid application performing numerical modeling in support of postwar atomic bomb projects. For the first time, calculations which

had previously required specialized mechanical hardware could now be performed on the same general computer. This was revolutionary, and a variety of numyerical schemes appeared in the early years of the computer age to take advantage of this newfound power, such as the Monte Carlo and Finite Element methods (Metropolis, 1987). With the rapid increase in computing power that has taken place since ENIAC first went online in 1946, it finally become possible to numerically investigate nonlinear systems; no longer are they only the concern of experimentalists (Wolfram, 2003). Following Moore's law, computing power has increased reliably and exponentially since the 1970s enabling more complex and more sophisticated computational modeling of even more nonlinear and complex phenomena. Modern numerical models range from the simple to the highly complex depending on the real world system that is being modeled.

In many real world applications, the simple models are not only adequate, but in truth, ideal. Many real world applications can be at least approximated using simple models and linear dynamics, and such a model should obviously be used when the physical dynamics truly are simple and linear. However, it is common for a linear or other simplified model to be used in the derivation of control laws even when the totality of the dynamics are substantially more complex. A common approach along these lines is to perform a polynomial expansion (often but not always a linear expansion) about an operating point and restricting the use of the model to the vicinity of this point. Another option often used is parameter scheduling. The use of such simplifications are particularly evident in the field of model based control, as a much greater and more mature body of literature exists for linear control than nonlinear control theory. Many applications utilize linear control, due to its greater simplicity and more mature knowledge base, even when the underlying dynamics are highly nonlinear. Despite this, there are many real world applications where it is either nec-

essary or greatly beneficial to deal with nonlinear models, rather than attempting to approximate around them. There are many real world applications where the benefits of involving a more complex, nonlinear model are worth it.

The truly massive amount of computing power available today has brought changes in modeling decision making – sophisticated, complex, and (person) time consuming analysis which might otherwise be necessary can often be replaced by brute computational force. More and more scientists make the decision to simply let the model fully resolve the complex nonlinear dynamics of the system, rather than attempting to derive an often time consuming approximation which might produce acceptable results. In the past there was often no choice to be made, the approximation could be solved, while the true model could not. A classical and straightforward example of this (although hardly the only one) comes from the world of computational fluid dynamics, specifically modeling. A great deal of effort has historically gone into turbulence models to approximate the effects of chaotic turbulence (Chen & Jaw, 1998, Rebollo & Lewandowski, 2014). The motivation for this has been that fully resolving all the length scales of turbulence in most flows would require extremely fine mesh sizes, resulting in impractically large problems. However, as computational power increases, we are slowly beginning to see the use of DNS for turbulent flow simulations in smaller scale, lower Reynolds number scenarios. An even simpler example is the transition over time from potential flow based simulations to simulations based upon Reynolds Averaged Navier Stokes (RANS).

The complexity of a modeled system can also affect the difficulty associated with validating the model. The classical example here is the modeling of chaotic systems. Regardless of whether they are time series or spatiotemporal, the simulation of chaotic phenomena necessarily involves more complex models (at least in terms of the model

dynamics, if not model form) than applications which are devoid of chaos. Referencing the previous example of turbulent flow, accurate simulation of turbulent pipe flow requires a substantially more complex model than the simulation of laminar flow (which is not chaotic) over an identical computational domain (Chen & Jaw, 1998, Rebollo & Lewandowski, 2014). Since the dynamics are chaotic, small changes in the initial condition (by definition) may produce large changes in the observed solution. However, since the model is chaotic the same could be said for difference between the model and reality. In other words, small errors in the model can also produce large differences between the model solution and experimental observations. Validating a model under these circumstances is clearly a much more complex and difficult task than it would be for a simpler system which does not exhibit chaotic behavior.

The complexities and difficulties associated with the modeling and validation of chaotic dynamics are relevant to the field of ship dynamics. One of the most striking arguments that can be made regarding this is to simply consider the form of analytical models of dynamic ship behavior. One such model is that developed by Soliman & Thompson (1991):

$$\ddot{\theta} + b_1\dot{\theta} + b_2|\dot{\theta}|\dot{\theta} + c_1\theta + c_2|\theta|\theta + c_3\theta^3 + c_4|\theta|\theta^3 + c_5\theta^5 = \frac{M(t)}{I}\frac{W_M}{I}$$

This is in truth a fairly simplistic model, yet it nevertheless contains a great deal of complexity and nonlinearity. Despite its complexity, it considers only a single degree of freedom (roll) and has constant coefficient values. One will also note that the model makes no consideration of coupling between degrees of freedom. Such coupling is real and only adds complexity and nonlinearity. Indeed, one of the most discussed nonlinear ship phenomena of recent years (parametric roll) is the direct result of such coupling.

A commonly accepted mathematical indicator of the presence of chaos is a positive maximum Lyapunov exponent. This has been observed in both experimental and numerical ship data. Early work in this area was done by Papoulias and Falzarano in the late 1980s and early 1990s. Papoulias (Papoulias, 1987) used a three degree of freedom model to investigate a mooring system for tankers, and used Lyapunov exponents to confirm the existence of chaotic and unstable behavior. Falzarano (Falzarano, 1992) investigated the capsize of fishing vessels, and demonstrated that positive maximum Lyapunov exponent values being associated with chaotic capsize behavior of the vessel.

Additional work has taken place in more recent years. Spyrou investigated large amplitude motions in quartering waves, with positive maximum Lyapunov exponent values being used to detect boundaries between stable and chaotic behaviors (Spyrou, 1996). Arnold investigated ship dynamics models, and found capsize to represent a bifurcation with a stable periodic orbit with a negative maximum Lyapunov exponent transitioning to chaotic behavior with a positive maximum Lyapunov exponent (Arnold, 2003). A series of works in the 2000s by McCue et al demonstrated that finite time Lyapunov exponents can be used to predict capsize behavior before it occurs in both experimental data sets as well as in simulations (McCue, 2004, McCue, 2006, Story, 2009). The total history of this work, and especially given the fact that positive Lyapunov exponents can be used to predict capsize, provides strong evidence that capsize is often the result of chaotic dynamics. This makes the accurate simulation of chaotic dynamics of critical importance to the fields of ship dynamics and ocean engineering.

In addition to exhibiting chaotic dynamics, ship capsize simulations present some special challenges. By their chaotic nature, they are difficult to validate, since the ac-

tual trajectory can be substantially changed by not only errors in the physics model, but also simply by small changes in the initial condition (even if the model is completely accurate). To some extent this can be mitigated by sampling the entire set of physically meaningful initial conditions when performing ship dynamics simulations. Ships can usually be modeled as a rigid body, meaning its state can be defined by a (multivariate) time series, rather than a more expensive spatiotemporal field. However, the behavior of a ship is heavily influenced by the surrounding hydrodynamics, which are not so easily modeled. As a result, while the output of a simulation might be a simple time series, the complexity of the simulation is akin to a spatiotemporal simulation, not a time series simulation.

For this reason, high fidelity simulations of ship dynamics are generally quite expensive numerically, even when substantial assumptions are made. This limits our ability to attack the problems associated with the chaotic nature of ship dynamics with high sampling counts. Instead, a two-prong approach is suggested. To some extent, sampling of initial conditions will always be necessary, to explore questions such as the probability of capsize occurring under a particular set of conditions. Given the computational cost of each individual simulation, however, it is important to develop techniques which allow the necessary answers to be generated using as few samples as possible there are problems amenable to ten thousand sample Monte-Carlo analyses, but this is not generally one of them.

An additional consequence of computational cost of ship dynamics simulations is that sophisticated approaches are needed to address the validation of ship dynamics simulations. While it is bad enough paying a heavy computational cost deriving answers from a computational model, paying a heavy cost to test whether a model matches experimental data is even worse, adding to an already high computational

investment. Not only is time lost performing the validation simulations, but there is no guarantee that the validation will be successful. The discovery of a small, easily fixed error in the model may take far longer to discover than it would to fix. Even in the case of physics which are conducive to easily solved models, performing large numbers of validation experiments (as would be needed to compare probabilistically with simulations sampled over the initial condition space) is almost always expensive. Both of these issues can be sidestepped by the development of validation methods which can reliably compare simulations and experiments based upon a small number of trajectories from each.

The majority of this body of work focuses on addressing the limitations discussed above. There are three separate areas of contribution made to the process of making a model useful in the real world. All relate to better enabling real world applications of complex nonlinear models. The first is validation, as previously mentioned, validation of a model is critical in proving that the model correctly represents the physics that exist in the real world. The second is the area of uncertainty quantification, which is useful in understanding how much error is introduced to the system by the algorithms used to solve it. The third area is control, which is directly using the validated model and applying control theory to make the model useful in the real world. The first three chapters following the introduction chapter are related directly to the area of validation; while the fourth is related to uncertainty quantification, and the last is applying control theory to a real world problems.

Chapter 2: "Effectiveness of chaotic system measures for the validation of ship dynamics simulations" is the first step in the development of a suite of validation tools for the validation of chaotic time series data. As previously mentioned, chaotic time series data is unlike regular time series data in that direct comparison is not an op-

tion for the validation of chaotic results. The sensitivity to initial conditions that is fundamental to chaos prevents the use of a direct comparison between two time series to determine the accuracy of a model. Two very different time series could still represent the same physics but with a direct comparison appear completely disparate. The tools proposed in this work are dynamical analysis tools that seek to quantify physical properties of a chaotic system. In this work, the tools are tested against canonical systems to determine their sensitivity to the changing dynamics of a system, specifically bifurcations. All methods tested showed promise and could, with further testing, be used as validation tools for chaotic time series data.

Chapter 3: "On the Use of Nonlinear System Tools to Validate Complex Ship Dynamics Simulations" is the second and final step in the development of the chaotic time series validation tools. This chapter includes testing on the same metrics presented in the previous chapter. Rather than continuing with canonical systems, a commercial six degree of freedom ship motion code was used to simulate a data set (Hayden, 2006). The code used has four different code variants (linear, small-approximate, large-approximate, and nonlinear). The linear code variant includes only a linear approximation of ship behavior. The small-approximate option allows for some nonlinearities to be accounted for within the confines of an assumption that sway motions are small. The large-approximate option is the same as small-approximate with the additional ability to handle large motions in sway. Lastly, the nonlinear option represents a fully nonlinear simulation that is comparatively expensive computationally.

Simulations were performed with varying levels of model fidelity (e.g. the four code variants); five data sets were run with a modest number of initial conditions for all four code variants. A much larger number of data sets were run with double the initial conditions but for only the first three code variants due to time constraints,

as the fully nonlinear code takes significantly longer to run with this particular data set. The metrics were calculated on each of these simulation types for the selected dataset in two ways; for the first metric calculation only roll and pitch information was given and for the second calculation roll, pitch and their corresponding velocities were given. Some metrics performed better than others and it is presumed that these metrics will perform better for some data sets than others. Better performance was defined as increased sensitivity to the physics as opposed to sensitivity to the initial condition. These metrics have been proven to be useful as validation tools and directly contribute to the expressed purpose of this body of work, to bring models to full utility in the real world.

Chapter 4: "Experimental study on deformation of flexible wedge upon water entry" is a separate contribution to the validation field. More specifically, this experiment was designed to be used in the validation of SPH simulations modeling the deformation of a flexible wedge entry into water. A model wedge support was designed and built that allowed for a membrane to be wrapped around it. The constructed frame and membrane were then tested in a forced water entry experiment. A camera and grid system placed on the membrane allowed for the deformations in the membrane to be calculated. The experiment was kept as simple as possible with all key values and specific experimental parameters being recorded for the later use in SPH simulation validation (not a part of this body of work). Experiments are key in the validation of models as they can represent the real world in a more controlled environment, albeit simplified, it can still provide accurate data points to help refine models in preparation for their intended uses in the real world.

Chapter 5: "Non-intrusive polynomial chaos for efficient uncertainty analysis in parametric roll simulations" is the next chapter. This work focuses on the development

of a tool to estimate uncertainty and other statistical quantities of a system. This chapter presents two different parametric roll simulations utilizing the same ship motion program as in Chapter 3. A set of initial conditions were swept with a given set of parameters, resulting in a large set of simulations. Using these simulations, both Monte Carlo analysis and a collocation based implementation of non-intrusive polynomial chaos were separately performed. The same methods were performed on the alternated data set which only differed in the values for the mean ship speed and wave frequency, resulting in two different parametric roll cases. The first produced higher roll motion amplitudes while lower roll motion amplitudes were observed in the second. The errors of the NIPC were determined using the Monte Carlo results as the baseline (or true results). Results from the response surface method (RSM) were also considered briefly, and those errors calculated as well. Both the NIPC and RSM performed well for this data set in the absence of capsize. Previous studies have indicated that NIPC will not work well in the presence of bifurcations and results in this study confirmed that. However, in the absence of a bifurcation, NIPC presents a computationally efficient alternative to Monte Carlo with proper use. This is work provide yet another step towards the development of a model towards real world uses.

The final contribution to the body of work is Chapter 6: "Design of a controller for autonomous vessel recovery utilizing the prediction of host vessel motions". This chapter is, fittingly, the last and final step to utilizing a model in the real world by the addition of control theory. In this chapter, a set of control laws was developed to allow for the autonomous recovery of an unmanned surface vehicle (USV) to a host vessel using motion prediction information. The controllers were derived using a USV model developed by Sonnenburg (2012). The model was set up in such a way as to allow for the decoupling of the controllers for different components of the system. Taking advantage of this property, there are two separate controllers, one

that controls the course and the other that controls the speed. The course controller is a relatively simple linear quadratic regulator (LQR), while the speed controller is a combination of backstepping and model reference adaptive control (MRAC). The controllers work together through a one way coupling, and guide the USV through two phases. The first phase is the waiting phase, which is where the USV trails the host ship at some desired distance. The second phase occurs when a trigger is given, then the USV is given an intercept course and will attempt to enter the host ship. Two types of triggers were evaluated for our system, the first was a neural network (NN) from Matlab's NN toolbox and the second is a nearest neighbor prediction approach developed by McCue (2008). Both predictors proved to provide sufficient warning to execute the entry maneuvers required by the developed control laws.

This body of work endeavors to contribute key pieces of information that are currently needed in the development of chaotic or highly nonlinear time series models for their uses in the real world.

CHAPTER II

Effectiveness of Chaotic System Measures for the Validation of Chaotic Canonical Systems

2.1 Introduction

Over the past few decades, the use of simulations has grown to the point where it has been referred to as the third pillar of science along with theory and experiments (Oberkampf and Roy 2010); this is particularly true in the ship dynamics community since experiments are difficult and costly to perform. Simulations provide a more manageable outlet to explore different scenarios which a ship might encounter. With these simulations comes the necessity of ensuring that such simulation is performed accurately, which is done through the processes of verification and validation. The main focus for the current study is on validation. For purposes of this work, the authors adopt the definition provided in the current version of the DoD Modeling and Simulation (M & S) Instruction (2009), which states:

> *[Validation is] the process of determining the degree to which a model or simulation and its associated data are an accurate representation of the*

This chapter is derived in large part from Cooper and McCue, 2012.

real world from the perspective of the intended uses of the model.

Validation has become well defined in field based simulations like Computational Fluid Dynamics (AIAA, 1998) and computational solid mechanics (ASME, 2006). For further readings see texts by Oberkampf and Roy (2010) and Roache (1998) as well as the DoD (2009) and DoN (1999) instructions. To say that one has validated a field based simulation has specific meaning due to well-defined methodologies in the above referenced manuscripts. In general, this is not the case with chaotic dynamic systems. Historically, visual inspection and later, statistical metrics were the primary means of validating simulation tools. These methods, while useful, leave room for improvement, especially when considering the nature of chaotic time series and their sensitivity to initial conditions. Yee et al. (1994, 1996, 1997a, 1997b) applied non-linear dynamics theories to computational fluid mechanics, and in a philosophically similar manner, Wu et al (2011) applied computational fluid mechanics research to code verification for chaotic systems. This chapter seeks to continue these past efforts towards improved validation tools drawing off expertise provided in the time series analysis and computational fluid dynamics research spheres.

The principle behind the use of chaotic systems measures is that they provide a quantitative metric that gives an indication into the underlying governing physics of the system. For example, while comparing capsize versus non-capsize for a given simulation/experiment pair may yield a false sense of security/insecurity as it is a binary metric; if the simulation and experiment have similar Lyapunov exponents or entropy or correlation dimension, one might be able to conclude both systems capture the same fundamental physics.

In recent years, several individuals have started to utilize metrics that have physical significance as validation tools (Bulian, 2005; McCue et al. 2006; McCue et al. 2008;

Reed, 2009). The work thus far is a positive step forward but there is a need to more fully characterize the capabilities of these physically based metrics, since previous works have primarily focused on using these metrics rather than understanding their role as a validation metric. In comparison to the usual approaches, these physically based metrics could serve to give more insight to the behavior of the system as well as potentially providing insight to why results may not match.

The current work will suggest several such metrics and provide a basic support for their use as a validation tool. Each metric is applied to two different well-known systems in order to aid in understanding the metric behavior. In addition, the metric results are evaluated using bifurcation analysis, which provides a coarse measure of the efficacy of the metrics. These steps provide an initial platform from which to continue the study of these metrics as a validation tool.

2.2 Methods

2.2.1 Dynamical Systems

Two different dynamical systems were used to test the metrics described in the next section. The first system, the Lorenz attractor, is a generic chaotic system, while the second system is a classic ship capsize model developed by Soliman and Thompson (1991). Both systems provide an excellent environment to test the following metrics due to their well characterized and understood behavior. The corresponding equations for each system can be seen in Table 2.1 along with the parameter values that were swept for each parameter. For both systems, 20 randomly generated initial conditions were used; the selection of the initial conditions was bounded roughly according to the size of the attractor for Lorenz, and just large enough to include variations but small enough to still generate stable trajectories for the Soliman and Thompson model

System	Equations	Parameters
Lorenz Attractor	$\dot{x} = \sigma(y - x)$ $\dot{y} = x(r - z) - y$ $\dot{z} = xy - \beta z$	$\sigma\,(0 - 20)$ $r\,(14 - 42)$ $\beta\left(0 - \frac{16}{3}\right)$
Soliman & Thompson Ship Capsize Model	$\ddot{\theta} + b_1\dot{\theta} + b_2\lvert\dot{\theta}\rvert\dot{\theta} + c_1\theta +$ $c_2\lvert\theta\rvert\theta + c_3\theta^3 + c_4\lvert\theta\rvert\theta^3 + c_5\theta^5$ $= \frac{M(t)}{I}\frac{W_M}{I}$	$\omega = 0.7, H(0.15 - 0.5)$ $\omega = 0.75, H(0.15 - 0.5)$ $\omega = 0.8, H(0.15 - 0.5)$

Table 2.1: Systems Studied

(Soliman and Thompson 1991). The Lorenz system simulations included a general sweep of parameters centered around the commonly referenced parameter values for the system (Lorenz α=10, β=8/3, r=28). For the Lorenz attractor, thirty evenly spaced parameter values within this range were run for each randomly generated initial condition. The Soliman and Thompson model was run with forty points within the range to increase resolution of the results.

2.2.2 Metrics

Correlation Dimension. The correlation dimension is commonly referred to as a type of fractal dimension and serves as a measure of the dimensionality of the space occupied by a set of points (Grassberger and Procaccia, 1983). The correlation dimension of a system can be calculated by:

$$C(\varepsilon) = \lim_{N\to\infty} \frac{g}{N^2} \sim \varepsilon^{\nu} \tag{2.1}$$

where $C(\epsilon)$ is the correlation integral, g is the total number of pairs of points with a distance less than ϵ, N is the total number of points, and ν is the correlation dimension.

Sample Entropy. Entropy is the measure of disorder or unpredictability of a given system. Note in this case entropy refers to information entropy rather than thermodynamic entropy. Sample entropy is a specific type of information entropy that is used for this work, the particular method of calculation used is defined by Lake et al (2002), who provided the code used in this study (http://physionet.incor.usp.br/physio tools/sampen/). Sample entropy can be calculated by finding sets of n points that are within some distance ε of each other, with sample entropy being defined as the negative logarithm of the probability that the $n+1$ point is within the set distance ε, so that:

$$S(\varepsilon, N) = -\log(p(x_N \in \varepsilon | x_{1,\ldots,N-1} \in \varepsilon)) \tag{2.2}$$

Proper Orthogonal Decomoposition. Proper orthogonal decomposition, also known as principal components analysis or the Karhunen-Loeve transform, transforms a set of correlated variables into a set of linearly independent variables so that the original data can represented as:

$$\overrightarrow{x}_i = \sum_{k=1}^{N} \lambda_{ik} \overrightarrow{v}_k \tag{2.3}$$

where x represents the vector data points, v represents linearly independent modes, and λ represents projection coefficients weighting each model to obtain a given data point. POD is unique in that the vectors v are chosen in such a way so that each successive mode accounts for the largest possible variance (Pearson, 1901). Thus, POD gives a simplified view of the data set which contains as much of the energy as possible. This is demonstrated graphically in Figure 2.1. For the purposes of this thesis, trajectories were compared by taking the dot product of their most energetic modes, as calculated by POD. Trajectories on identical attractors should have identical first modes, and hence would have a validation score of 1, with more distant trajectories having lower scores.

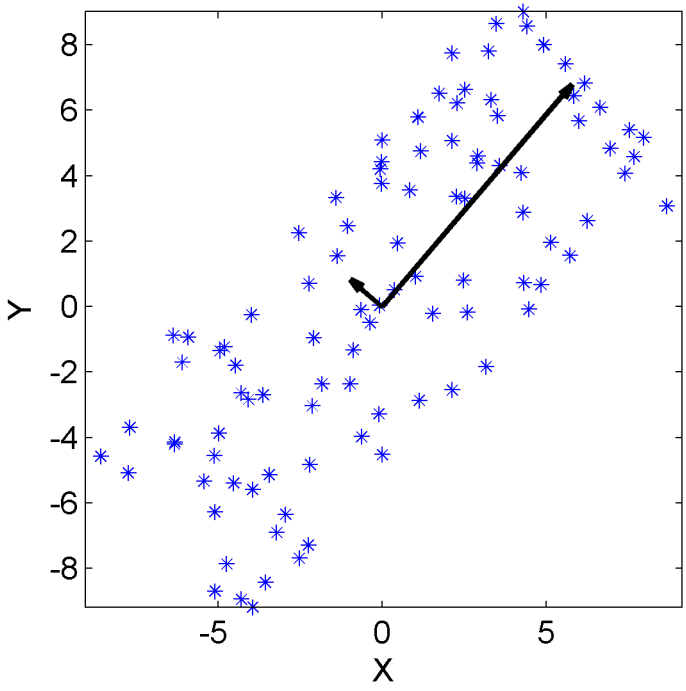


Figure 2.1: Example of proper orthogonal decomposition. The blue points represent scattered data, while the black arrows represent the first and second modes (with the longer arrow corresponding to the first mode). Note how the first mode is aligned with the largest spread of the data.

Hurst Exponent. The Hurst exponent was developed to quantify the degree to which a system exhibits long-term memory. It is classicaly defined as:

$$\lim_{n\to\infty} E\left[\frac{R(n)}{S(n)}\right] = Cn^H \tag{2.4}$$

where R represents the rescaled range of the cumulative deviate of a subset of the data and S represents the standard deviation of the same. n represents the number of samples included in a subset. Note that the cumulative deviate is defined as:

$$C_i = \sum_{i=1:n} x_i - \mu \tag{2.5}$$

The physical meaning of the exponent is as follows. All values of the Hurst exponent lie on the interval [0,1]. Values above 0.5 indicate correlation over long time scales, while values greater than 0.5 indicate a time series which switches comparatively between states above and below the mean. Nominally, a value of 0.5 indicates a lack of correlation, but may also under some very specialized circumstances with correlated time series.

In principle, the method discussed above is only applicable to scalar time series; there is debate regarding the appropriate methodology for calculating the multivariate Hurst exponent. Here, we allow for multivariate time series by substituting a vector form of the cumulative deviate described in Eq. 2.5 by substituting the following vector alternative:

$$C_i = \left| \sum_{i=1:n} \overrightarrow{x_i} - \overrightarrow{\mu} \right| \tag{2.6}$$

Lyapunov Exponent. The well known Lyapunov exponent characterizes the rate at which nearby trajectories diverge, and is often used to characterize the degree to which a system exhibits chaotic behavior. It is normally defined as:

$$|d\mathbf{Z}(t)| = e^{\lambda t} |d\mathbf{Z}(0)| \tag{2.7}$$

In general, $\mathbf{Z}$ represents the vector state of a multidimensional system. In this case, there is not a single Lyapunov exponent, but rather a spectrum of them (one per system dimension). Each exponent quantifies the contraction or divergence of the system in different directions. Note that these are not reported in the system coordinates, but rather in a coordinate system aligned with the direction of maximum divergence. Thus, the first Lyapunov exponent has special meaning, in that it quantifies the total rate of divergence between nearby trajectories and is often referred to as the Maximal Lyapunov Exponent (MLE). Positive values for this exponent are a key indicator

of chaotic system behavior. Moreover, it is common practice to calculate a single Lyapunov spectrum for a given attractor shape, treating the Lyapunov spectrum as a property of the attractor. This makes the MLE, or the Lyapunov spectrum as a whole, attractive as potential validation metrics for time series simulations. Indeed, as discussed previously, several author's have used it for such purposes, although an investigation of it's validity as such was not performed. In this work, only the MLE as a scalar is considered, rather than the Lyapunov spectrum as a whole.

There are two separate and distinct methods for calculating the Lyapunov exponent. It is common to calculate Lyapunov exponents by simultaneously simulating an expanded system, where the trajectories of linearly perturbed points are simulated simultaneously with the trajectory of the central, main point (Strogatz, 1994). This approach generally produces excellent results, and is viable when an analytical model exists. However, when a model cannot be expressed as analytic equations, or for experimental data, such an approach cannot be used, and Lyapunov exponents must be calculated directly from time series data. Several approaches for such calculations exist; herein, we use that proposed by Sano and Sawada (1985). For the two systems considered here, the Lyapunov exponents for the Lorenz attractor are calculated using the extended system approach, while Lyapunov exponents for the Soliman and Thompson ship capsize model are calculated from the time series data.

It should be noted that many of these approaches have already been discussed with application to experimental data in the literature; see, for example, McCue et al (2008) which uses recurrence plots, correlation integral, correlation dimension, Poincare sampling, and attractor reconstruction as qualitative and quantitative metrics for comparing numerical simulations to experimental data. The purpose of the present chapter is more fundamental and in line with the bifurcation study for Lyapunov exponents

presented in Story et al (2009, 2010). Specifically, the authors seek to demonstrate that the quantitative metrics used in this dissertation correlate to results of bifurcation analysis to determine if the metrics indeed capture changes in fundamental physical behavior.

2.3 Results and Discussion

As previously mentioned, the metrics laid out in the prior section were tested on two different systems, the Lorenz attractor and the Soliman and Thompson capsize model.

2.3.1 Lorenz Attractor

Each parameter was varied in this system according to the methods describe in Section 2.2.1. The results of each parameter variation (σ,r,β) are presented in a single figure for each metric tested. In each figure there are the results with error bars of $\pm$ one standard deviation at each data point, as well as vertical lines that represents a given bifurcation point. The bifurcation values were generated by Story (2009) using AUTO.

Correlation Dimension. The results for correlation dimension can be seen in Figure 2.2. For each of the parameter variations there appears to be a relatively close correlation between trend changes in the correlation dimension value and the bifurcations. Another point of interest is that this metric is fairly robust, with respect to initial conditions, as indicated by the small error bars seen in all three parameter variations particularly when distant from a bifurcation. In Figure 2.2, for small values of σ, correlation dimension values are also very small, which is due to the fact that the trajectories quickly tend towards the fixed point, which represents a 0-D geometry.

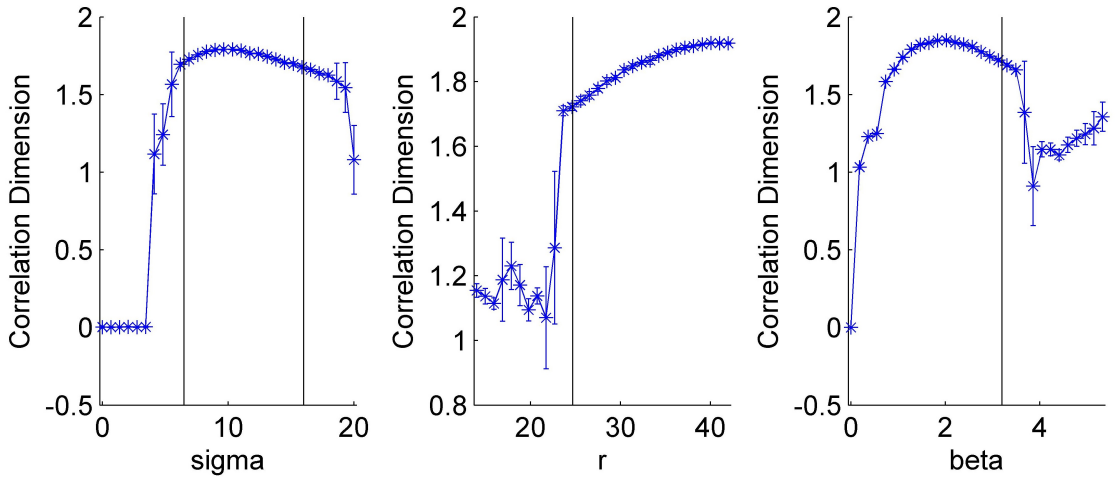


Figure 2.2: Parameter sweep for correlation dimension - Lorenz. Vertical black lines represent bifurcation values.

As the attractor shape becomes more defined, with increasing values, the correlation dimension increases as well. Similar behavior can be seen in the r value changes as well, but the difference is that for small values of r, the correlation dimension is larger than the values for σ because the trajectories do not move as quickly to the fixed point. In addition, the β variations develop similarly to the r variations except that the attractor becomes fully developed more quickly, as seen by the quickly rising correlation dimension values.

Sample Entropy. The sample entropy results for the Lorenz attractor are shown in Figure 2.3. These results are quite similar to the correlation dimension results shown in Figure 2.2. As with the correlation dimensions results, the sample entropy results show trend changes in relatively close proximity to the bifurcations, indicating that this method does correlate to changes in shape/behavior of the attractor. In addition, the error bars are very small for all three parameter variations, indicating this method is robust to initial conditions. Also, small sample entropy values can be seen for small values of σ, which is a similar behavior to that observed with correlation

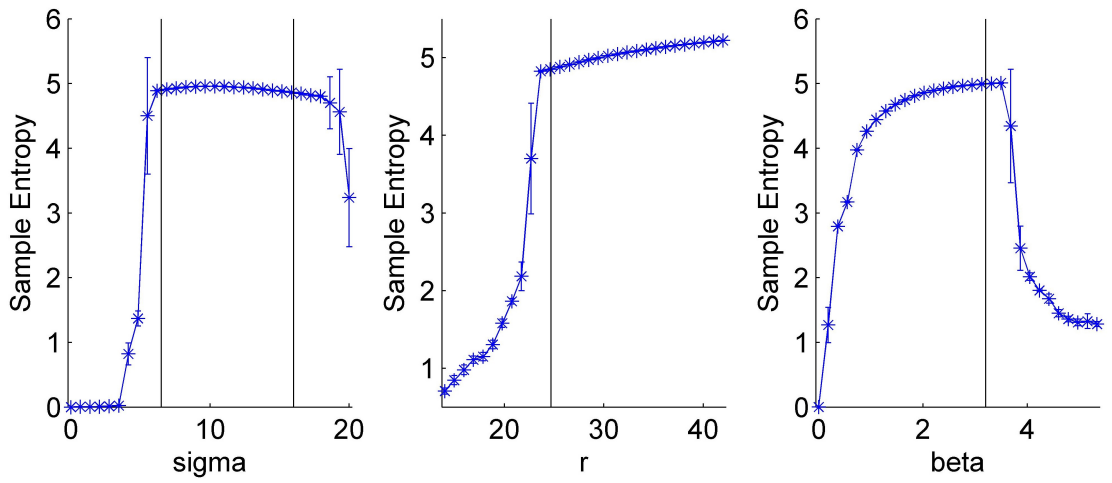


Figure 2.3: Parameter sweep for sample entropy - Lorenz. Vertical black lines represent the canonical bifurcation values.

dimension. The same trends can be seen in the r and β variations for sample entropy as well.

POD. The results for the POD metric can be seen in Figure 2.4. The authors note the error bars are larger with this metric than with many of the other metrics. For the σ and r variations, the error bars appear before the bifurcation, which means that there is a larger standard deviation in the results before the attractor is fully developed. This is opposite the behavior observed with β variations; in this case the attractor develops quickly and starts to degrade after the bifurcation, which is where the larger error bars appear. In general the results tend to change near the bifurcation, slightly trending up or down in the cases of σ and β and more sharply with the r variation.

Hurst Exponent. The behavior of the Hurst exponent is rather unique in comparison to that of the other metrics; these results can be seen in Figure 2.5. It does not necessarily demonstrate strong changes in value across bifurcations, however, there are significant changes in the metric value as the bifurcation occurs. It is not clear

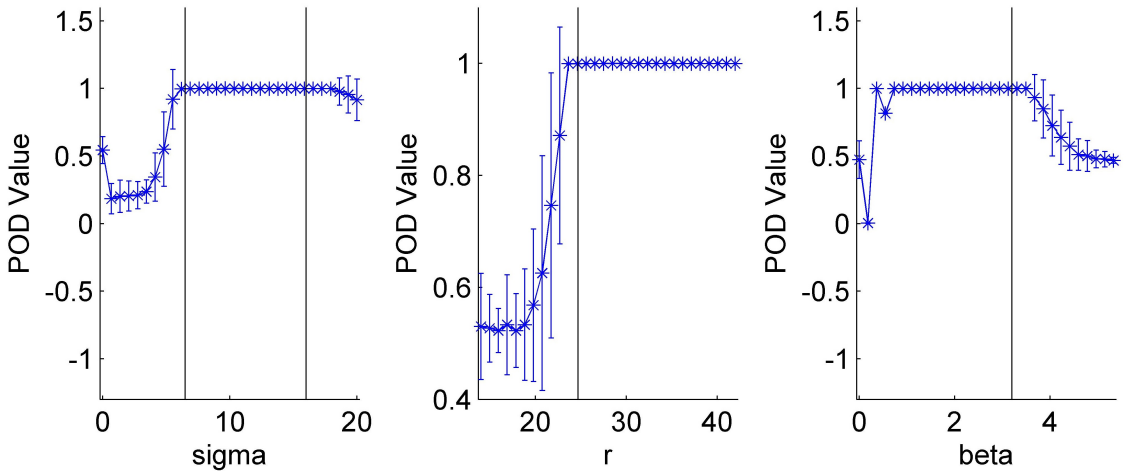


Figure 2.4: Parameter sweep for POD - Lorenz. Vertical black lines represent the canonical bifurcation values.

why this occurs. It is also important to note that the error bars are substantially larger for the Hurst exponent in comparison to other candidate metrics such as the sample entropy. Although this spread is not desirable in a validation metric for chaotic systems, the difference in the behavior of the Hurst Exponent compared to the other metrics considered suggests that it may provide insight distinct from that provided by other metrics.

Lyapunov Exponent. The results for the Lyapunov exponent are shown in Figure 2.6. These results show both strong changes associated with the bifurcations as well as negligible sensitivity to the initial condition. These results suggest that, at least when calculated using the extended system method, the Lyapunov exponent represents an excellent validation metric.

2.3.2 Soliman and Thompson Model

For the Soliman and Thompson model, the parameter ω, which is the ratio of the forcing frequency to the linear natural frequency, was given three different values,

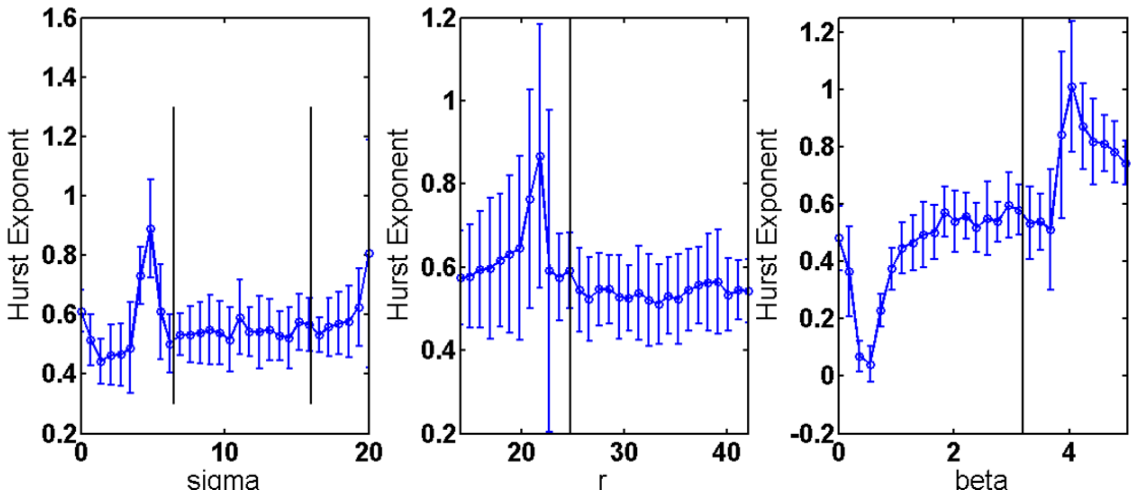


Figure 2.5: Parameter sweep for the Hurst Exponent - Lorenz. Vertical black lines represent the canonical bifurcation values.

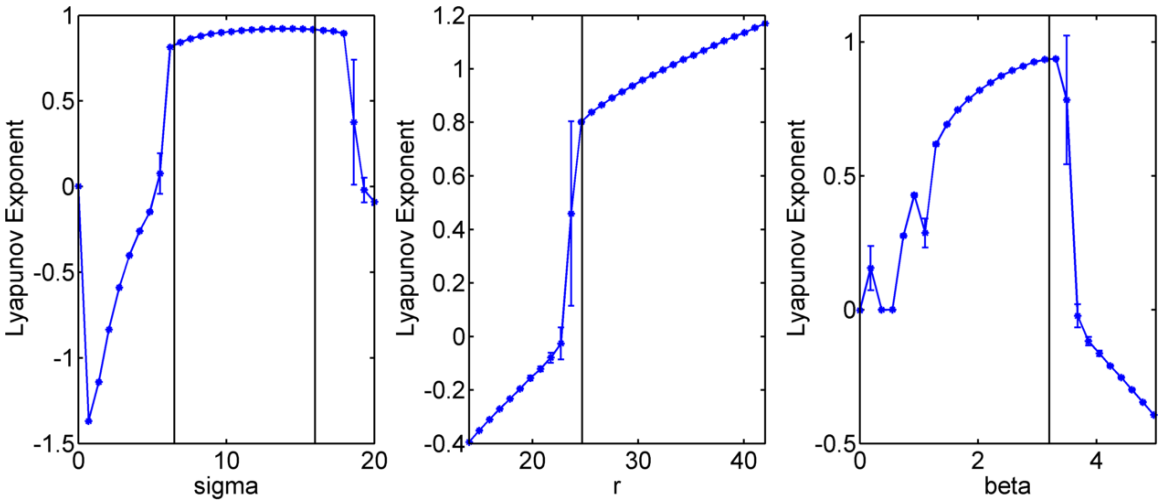


Figure 2.6: Parameter sweep for the Lyapunov Exponent - Lorenz. Vertical black lines represent the canonical bifurcation values.

0.7, 0.75, and 0.8 while the wave height parameter, H, was varied from 0.15 to 0.5. Corresponding figures are presented in the same manner as described in Section 2.3.1 for the Lorenz attractor. The model was run with the *Gaul* parameter values and the bifurcation values were taken from Table 2.1 in Soliman and Thompson (1991).

Correlation Dimension. The results for correlation dimension are shown in Figure 2.7. It is immediately apparent that there are definite and clear changes in behavior near the bifurcations for all values of ω. Other changes of note are the slight to a sometimes large jump in correlation dimension value and that after the bifurcation the error bars become quite large. The large error bars occur due to the nature of this model, prior to the bifurcation the trajectories are stable, rotating around a fixed point, whereas after the bifurcation the trajectories branch off toward infinity, physically representing capsize. The correlation dimension is very sensitive to the amount of time spent populating the defined phase space prior to diverging to infinity and therefore results in a larger spread of data (i.e. higher standard deviation and larger error bars). For the first case, ω=0.7, until the wave height parameter, H, gets to

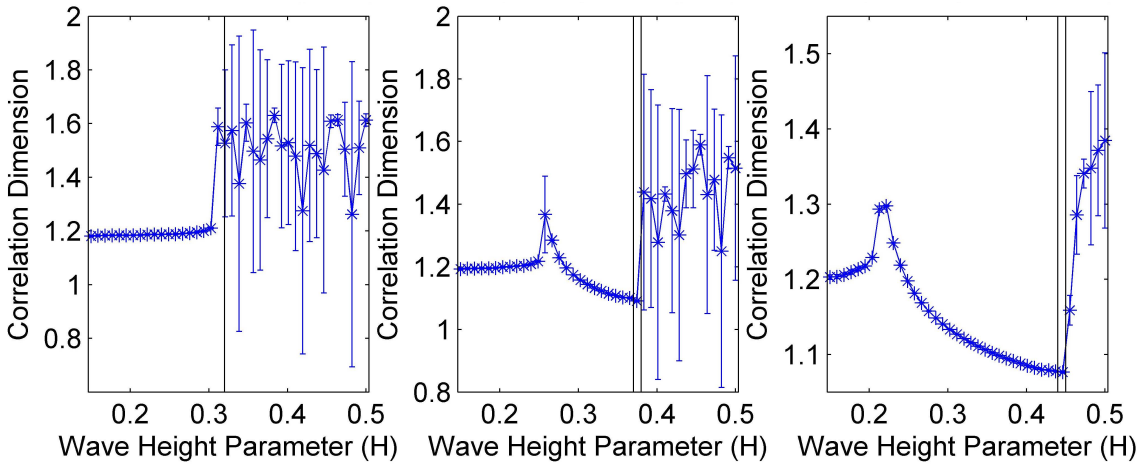


Figure 2.7: Parameter sweep on H for ω=0.7 (left), 0.75 (middle), and 0.8 (right) for correlation dimension - S&T. Vertical black lines represent bifurcation values.

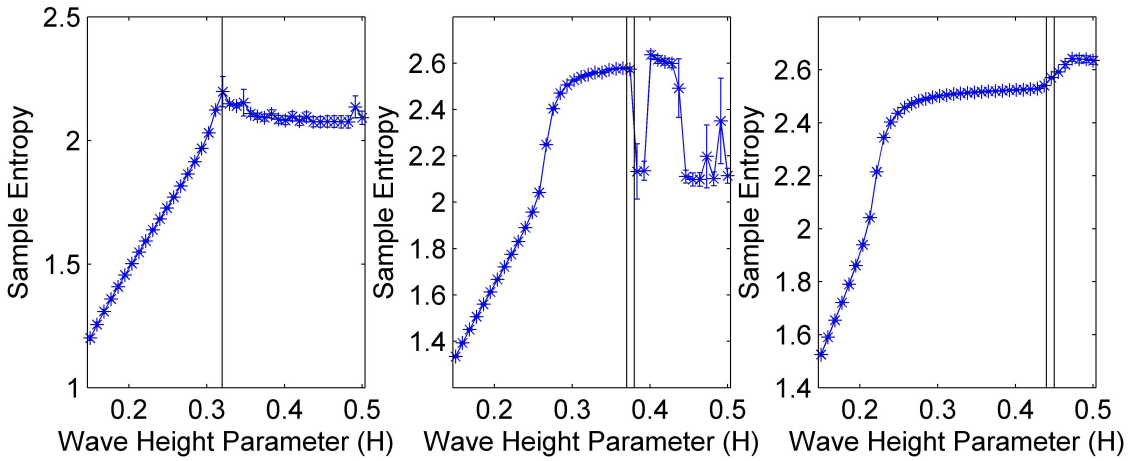


Figure 2.8: Parameter sweep on H for ω=0.7 (left), 0.75 (middle), and 0.8 (right) for sample entropy - S&T. Vertical black lines represent bifurcation values.

approximately 0.3, the correlation dimension is relatively stable at a value of 1.2, but after the bifurcation the values change drastically due to this phase space sensitivity. The same trend is seen in the second case, ω=0.75, with the exception that there is a small peak in the data around H=0.26, which occurs due to a slight growth in the size of the attractor, which then decreases in size slightly as it approaches the bifurcation. The third case, ω=0.8, is very much the same as with the second case, with a slight peak occurring around H=0.22.

Sample Entropy. Figure 2.8 shows the results for Sample Entropy; the primary similarity between the three ω values is that the error bars are very small after the bifurcation and very small prior to bifurcation. For the first case (ω=0.7), there is a gradual upward trend towards the bifurcation, where the results plateau. The second case (ω=0.75) is different because there is a slight drop in the sample entropy value around the bifurcations, but this value increases almost immediately after the bifurcation then decreases again to a plateau. These changes track very well with the changes in the space that the attractor occupies, which corresponds well to what

entropy physically represents: a measure of disorder of a system. For the last case (ω=0.8), there is a gradual upward trend in value overall but no clear change in behavior around the bifurcations.

POD.

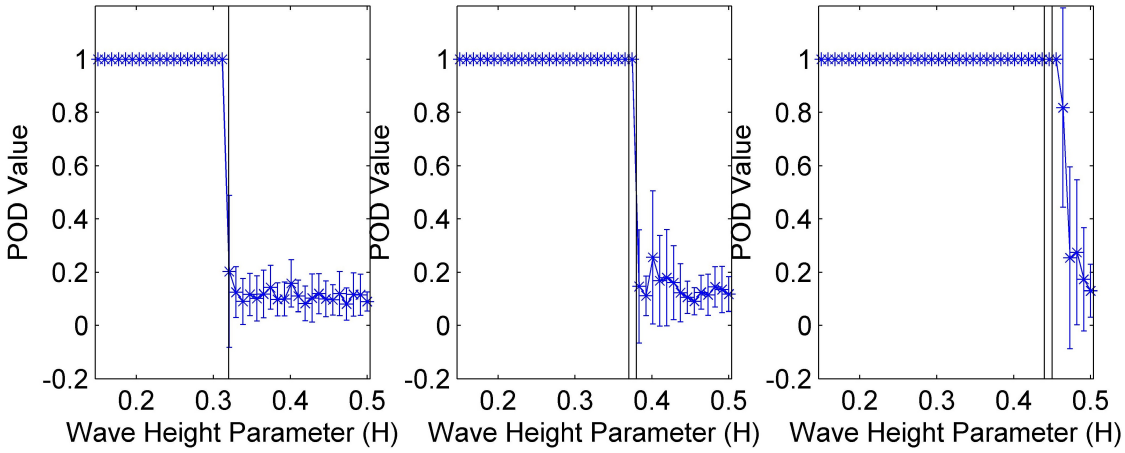


Figure 2.9: Parameter sweep on H for ω=0.7 (left), 0.75 (middle), and 0.8 (right) for POD - S&T. Vertical black lines represent bifurcation values.

The POD results in Figure 2.9 show strong changes near the bifurcation for all three values of ω. This occurs because the limit cycle dominates the first mode for all parameter values before the bifurcation. After the bifurcation, however, the escape trajectory dominates the first mode. This accounts both for the changes in the first mode as well as the large increase in variability observed after the bifurcation, as the trajectories escape towards infinity in a variety of directions.

Hurst Exponent.

The results for the Hurst Exponent are shown in 2.10. It is interesting to note that the Hurst exponent seems to perform substantially better for the Soliman and Thompson attractor than for the Lorenz attractor. It is not entirely clear why this is the case.

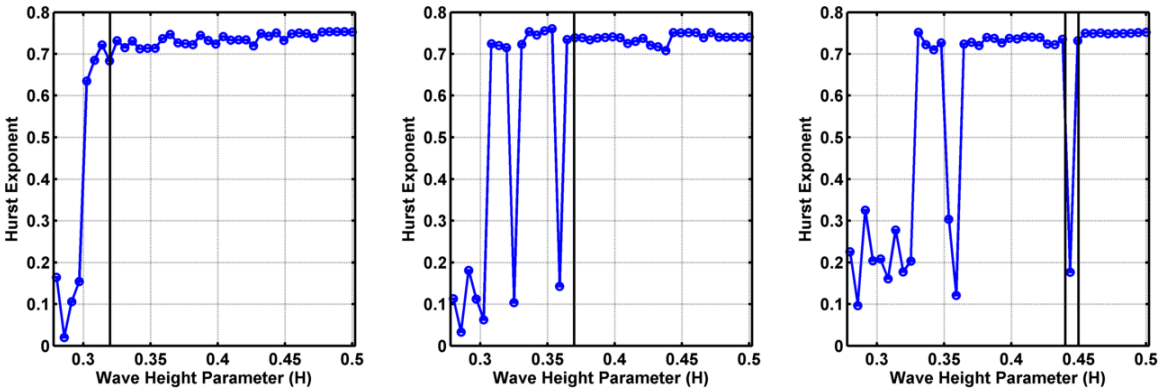


Figure 2.10: Parameter sweep on H for ω=0.7 (left), 0.75 (middle), and 0.8 (right) for the Hurst Exponent - S&T. Vertical black lines represent bifurcation values.

One potential difference is that the Soliman and Thompson is forced, while the Lorenz attractor is not, although this is pure spectulation. Regardless, the Hurst exponent clearly changes in value as the wave height parameter passes through the bifurcation, and has relatively small error bars. This is indicative of the behavior desired of a validation metric, and suggests that the Hurst Exponent has more potential as a validation metric than its previous application to the Lorenz attractor would suggest.

Lyapunov Exponent. Finally, the results for the Lyapunov exponent are shown in 2.11. Recall that unlike the results for the Lorenz attractor, the results here are calculated directly from the time series solution using the Sano and Sawada (1985) algorithm discussed previously. As was the case for the Lorenz attractor, the Lyapunov exponent demonstrates itself to be quite sensitive to changes in the behavior of the system. However, there is substantially increased variation between individual simulations compared to that observed compared to the previously shown Lorenz attractor results. This is attributed to the rate of convergence of Lyapunov exponents calculated directly from time series data, which is often substantially slower than that observed when using the extended system approach. This leads to an important point. If sufficient

data is present to calculate it, the Lyapunov exponent demonstates the potential to be an excellent validation metric. However, in many cases other metrics may perform better simply due to convergence issues. Keep in mind that Lyapunov exponents for experimental data will always need to calculated directly from time series data, and it is often more difficult to run an experiment for a longer period of time than it would be to continue a simulation for more time.

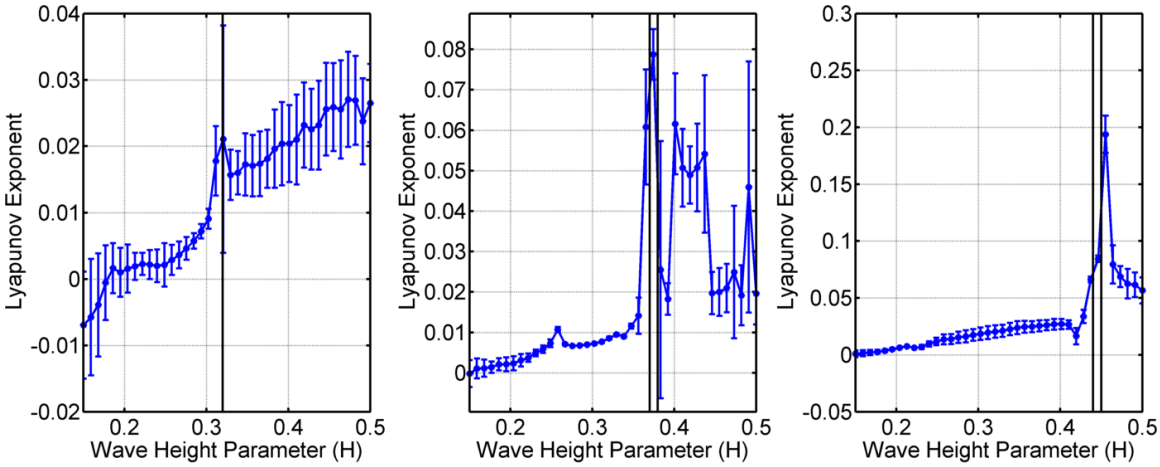


Figure 2.11: Parameter sweep on H for ω=0.7 (left), 0.75 (middle), and 0.8 (right) for the Lyapunov Exponent - S&T. Vertical black lines represent bifurcation values.

2.4 Conclusions

In this work, physically based measures of the properties of chaotic systems have been shown to be effective in detecting changes in system behavior. This suggests that these tools would be quite useful for the validation of chaotic time series simulations. However, this study investigated the behavior of candidate metrics for well understood canonical systems. For an initial study, this is ideal, as the expected behavior is well understood *a priori*. This represents an initial test regarding the viability of each candidate metric, one each metric passed by demonstrating an ability to detect

changes in system behavior. However, real world applications of validation inevitably involve numerical models whose behavior is not well understood *a priori*. Such an investigation is undertaken in Chapter 3 of this work; further conclusions regarding the relative advantages and disadvantages of the candidate metrics are drawn therein.

CHAPTER III

On the Use of Nonlinear System Tools to Validate Complex Ship Dynamics Simulations

3.1 Introduction

As discussed in Chapter 2, numerical simulations have particular importance in the ship dynamics community since experiments can be difficult and costly to perform. In ship motion modeling, chaos-like dynamics must often be considered. The validation of such simulations is still a relatively new field, and there have been has been a great dependence upon less than ideal methods such as direct comparison and simple statistics. These methods, while useful, leave room for improvement, especially when considering the nature of chaotic time series and specifically their sensitivity to initial conditions. This is in contrast to the validation of spatial simulations such as Computational Fluid Dynamics and Finite Element Analysis, for which well-established validation methods are defined (Oberkampf and Roy, 2010, Roach, 1998). The development of similar guidelines and methods for the simulation of chaotic time series phenomena would be beneficial.

In an effort to address this need, in Chapter 2 a series of dynamical system metrics were investigated as potential measures of simulation validation. These measures

were calculated for simulations of two canonical systems, the Lorenz attractor, and the Soliman and Thompson (1991) ship capsize model. There, simulations of the canonical systems were performed over a wide ranges of parameter values, with particular emphasis on where parameters passed through bifurcations, resulting in dramatic shifts in the behavior of the system. The goal of this study was to confirm that each candidate validation metric was able to clearly detect such strong changes in system dynamics. Each candidate metric considered in this study was deemed to have passed this preliminary test.

The current work further investigates the potential of these same metrics for use as validation tools. It should be noted that many of these approaches have already been discussed with application to experimental data in the literature; see, for example, McCue et al (2008) which uses recurrence plots, correlation integral, correlation dimension, Poincare sampling, and attractor reconstruction as qualitative and quantitative metrics for comparing numerical simulations to experimental data. The purpose of the present paper is more fundamental and in line with the bifurcation study for Lyapunov exponents presented in Story et al (2009, 2010). Specifically, the purpose is to investigate the sensitivity of the metrics to changes in physics and attractor topology, while remaining insensitive to trajectory changes associated with changes in the initial condition. This is evaluated by comparing calculated metric values for different simulations performed under itentical conditions but with different physics models, as discussed below. Unlike the previous study (Chapter 2), simulations by a state-of-the-art commercial ship motion code were used to investigate the utility of the candidate validation metrics. This provided a more rigorous test of the metrics, more akin to scenarios that would be expected in "real-world" applications.

3.2 Methods

3.2.1 Metrics

In this study, five separate measures of nonlinear time series behavior are investigated. These metrics are the correlation dimension, sample entropy, a comparison of principal orthogonal components (via principal orthogonal decomposition), the the Hurst exponent, and finally the Lyapunov exponent. These are the same metrics considered in Chapter 2, where each metric is individually described. In the interest of avoiding redundancies, such descriptions are not repeated here.

3.2.2 Numerical Simulations

To examine the behavior of the individual validation metrics, simulations were performed using the DTMB Model 5514 hull form, which represents a notional destroyer model (Hayden, 2006). These simulations were performed using a commercial, state-of-the-art multi-degree of freedom ship motion code. This code has four different versions, which are "linear," "approximate-small," "approximate-large," and "nonlinear," and which correspond to increasingly complex physics models. The fully nonlinear code version is substantially more computationally expensive compared to the other three, and is generally used for model validation.

These simulations were modeled after a series of experiments which were performed at the David Tayor Model Basin, Naval Surface Warfare Center, Carderock Division (Hayden et al, 2006). In these experiments, a scale version (1:46.6) of the same hull form was tested under a large number of dynamic seakeeping (among other) conditions, with wave frequency, wave height, and wave angle being varied. For the purposes of this chapter, two separate sets of data were considered; a small set of five data sets and also a large number of individual experiments (192) were considered

(the five are part of the larger data set).

For the smaller set of five wave conditions (for which nonlinear simulations were performed) the behavior and utility of the different candidate metrics is qualitatively investigated by inspection. For the larger set of simulations, however, the increased amount of available data allowed for more statistically rigorous conclusions to be drawn. Specifically, standard one-way Analysis of Variance (ANOVA) (Ott and Longnecker, 2001) was used to quantify the degree to which changes in candidate metric values are associated with changes in code version and simulated physics (as opposed to random effects). For an ideal validation metric, all variance present in the set of validation metric values should be due to changes in the physics, not random effects associated with changes in the initial condition (as opposed to different physics resulting from different initial conditions, which should change the metric value). In other words, every simulation performed for the same code version, experimental conditions, and physics should result in precisely the same validation metric value. The results of this analysis should make it clear which metrics best approximate this behavior.

3.3 Results & Discussion

For the first portion of this study, a smaller set of simulations were performed matching five different experimental wave conditions. These experimental conditions was selected based upon comparatively longer acquisition times (acquisition times are highly variable in the data set) and the absence of capsize. It is anticipated that these conditions should result in high quality calculations of the validation metrics for the experimental data, enabling comparison between simulated and experimental trajectories. The specific experiments included in this smaller set of simulations

were Run 213, Run 262, Run 285, Run 296, and Run 409. Separate simulations were performed for each of the code versions available within the simulation tool, with simulations being performed from thirty different initial conditions for each combination of wave condition and code version, so that there were 600 total simulations (four code variants x five wave conditions x 30 initial conditions). These initial conditions were varied randomly, with the initial roll and pitch angles varying between ± 10 degrees and roll and pitch velocities varying between ± 1 degrees per second. Additionally, the initial phase of the incident wave train was randomly generated. Note that all random samples were generated from uniform distributions.

3.3.1 Phase Portraits

To visualize the degree of similarity present between the simulated and experimental data sets for this smaller set of simulations, Figures 3.1-3.4 show phase portraits for in roll and pitch for the data associated with Run 213. In the interest of brevity, only the results associated with this experiment are discussed here; similar investigations were also performed for the other four sets of simulations, some of these can be seen in Appendix A. In several of the other simulations,there were significant numbers of simulations which capsized for the approximate-large simulations; the effect of these capsize simulations on metric calculations is discussed below. There are several important observations to make from the results from Run 213. First, it is apparent that while the phase portraits are not exact matches, there is a strong agreement between the simulated and experimental behaviors. The best agreement is observed with the fully nonlinear simulations shown in Figure 3.4, although even for these simulations there are some slight differences.

It is also important to consider the differences between the different simulations.

With the exception of the body linear simulations, similar pitch behavior is observed in the three remaining simulation types. There are also similarities in the roll behavior observed for both the approximate-small behavior and the fully nonlinear behavior (Figures 3.2 & 3.4, respectively). The roll behavior for both the linear and the approximate-large simulations are substantially different. However, the attractor shape associated with the approximate-small behavior is noticably wider than is the case for the fully simulation. Some of this difference in attractor shape could be accounted for by the difference in heading; the simulation heading was kept constant significantly better than the heading for the model tests, which often got "trapped" into a beam seas like condition. As a result, it is anticipated that there should be some differences in metric values between the results of each of the physics models.

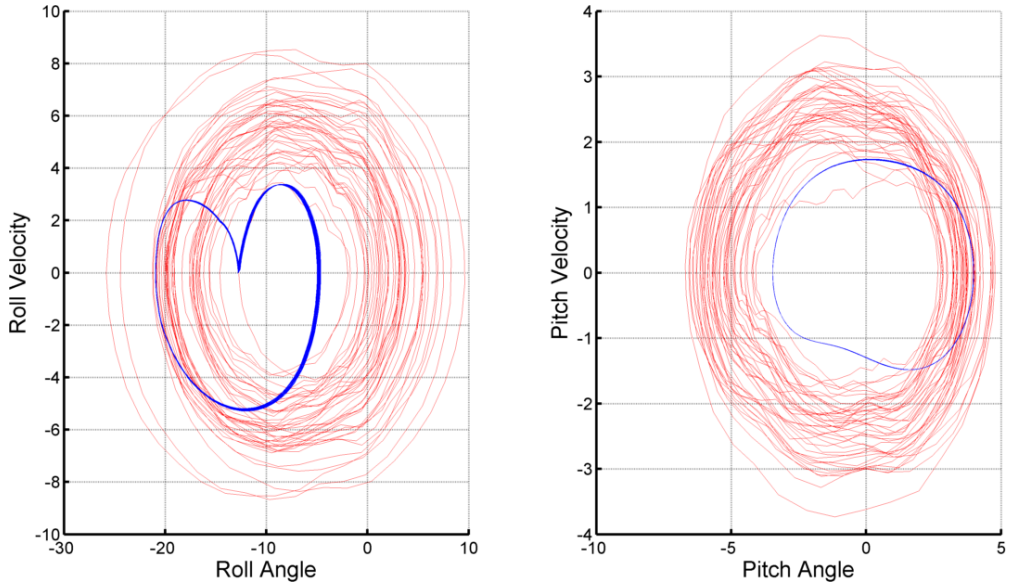


Figure 3.1: Comparison of experimental phase portraits with the phase portraits from the body linear simulation. Blue lines correspond to simualted data; red corresponds to experimental data. Data for experimental conditions matching Run 213.

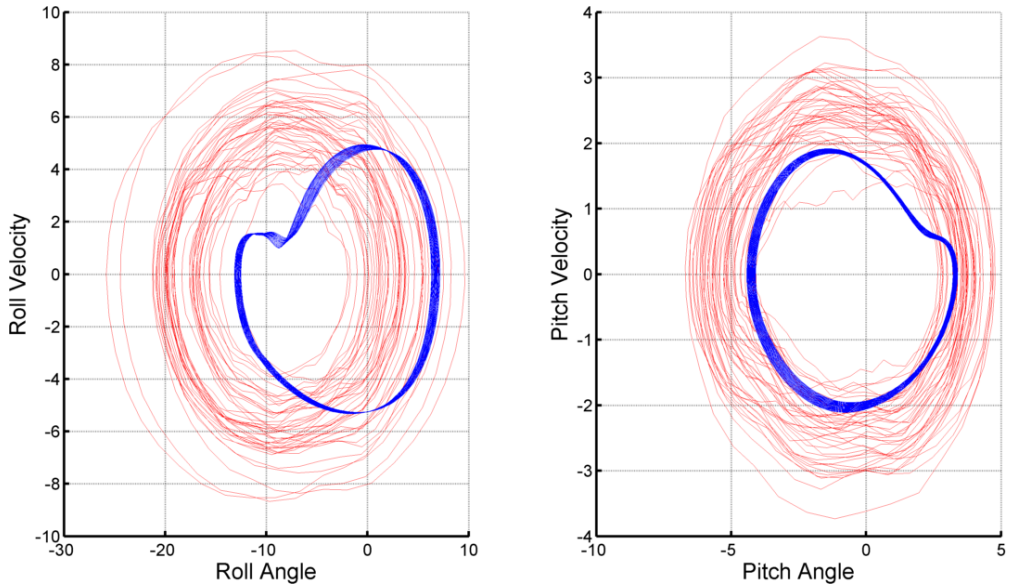


Figure 3.2: Comparison of experimental phase portraits with the phase portraits from the approximate-small simulation (approximately linear with small lateral motions). Blue lines correspond to simualted data; red corresponds to experimental data. Data for experimental conditions matching Run 213.

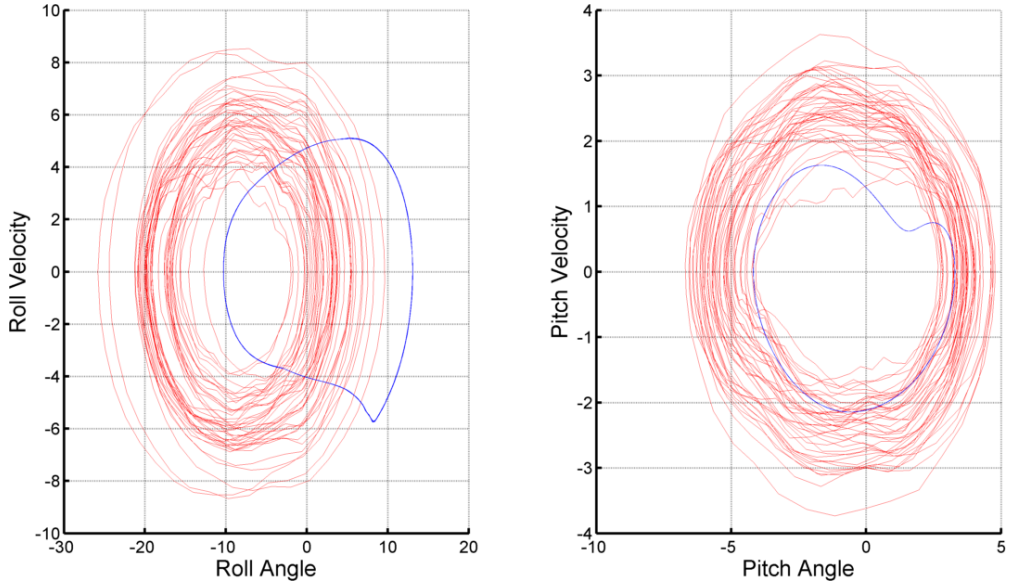


Figure 3.3: Comparison of experimental phase portraits with the phase portraits from the approximately-large simulation (approximately linear with large lateral motions). Blue lines correspond to simualted data; red corresponds to experimental data. Data for experimental conditions matching Run 213.

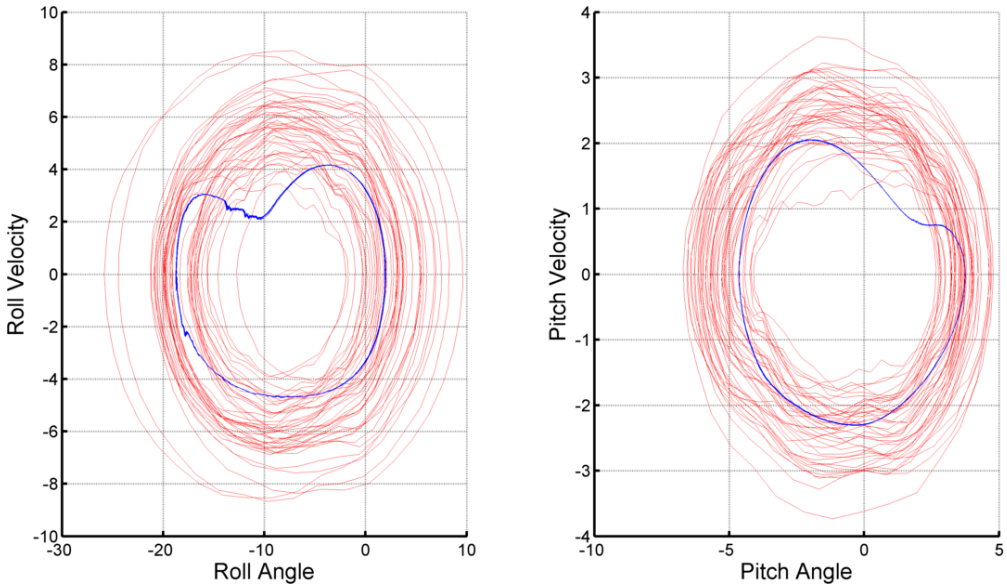


Figure 3.4: Comparison of experimental phase portraits with the phase portraits from the nonlinear simulation. Blue lines correspond to simulated data; red corresponds to experimental data. Data for experimental conditions matching Run 213.

3.3.2 Metric values - small data set

For each candidate validation metric, a population of values was calculated for each wave condition and code version. These populations were compared with the candidate validation metric values calculated from the experimental data. Separate comparisons were performed for 2-D and 4-D representations of the data. The data used for 2-D calculations consists of the roll and pitch angles as a function of time. Similarly, the 4-D data consists of the roll and pitch angles supplemented by the roll velocity and the pitch velocity.

The results for the correlation dimension are shown in Figure 3.5. Several important observations can be taken from these plots. First, it is noted that the values calculated for the correlation dimension appear to be unaffected by changes to the initial condition. The only exception to this is the results for approximate-large. However, inspection of the approximate-large simulation results showed a tendency towards

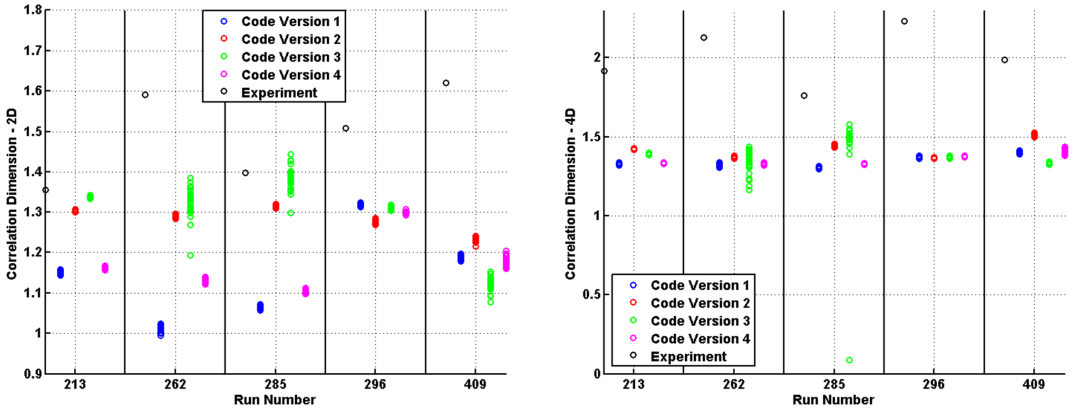


Figure 3.5: Calculated correlation dimension values for each simulated trajectory, as well as for each experiment *left:* calculated values based upon 2-D trajectory information *right:* calculated values based upon 4-D trajectory information. Note that with the exception of approximate-large, the variance among simulations associated with the same experiment is quite small compared to the differences between code versions.

capsize which was not observed in either the experiment nor in the other simulations (linear, approximate-small and fully nonlinear). Thus, the high levels of variability observed in the approximate-large data is attributed to variations in the actual simulations rather than to irrepeatabilities in the validation metrics. This observation applies to the data presented for each validation metric, and is most relevant to the data associated with Run 285, although capsize is observed in the data for other runs as well, albeit to a lesser extent.

Having satisfied that the correlation dimension seems to be negligibly affected by the initial condition, the next consideration is whether metric values are affected by the inclusion of different levels of physics into the model. In other words, are significant differences in the metric values calculated from different code versions? By examining the results shown in Figure 3.5, we see that this seems to be the case. The combination of these two factors (low sensitivity to initial condition and significant sensitivity to changes in physics) is indicative of a useful and valid validation metric.

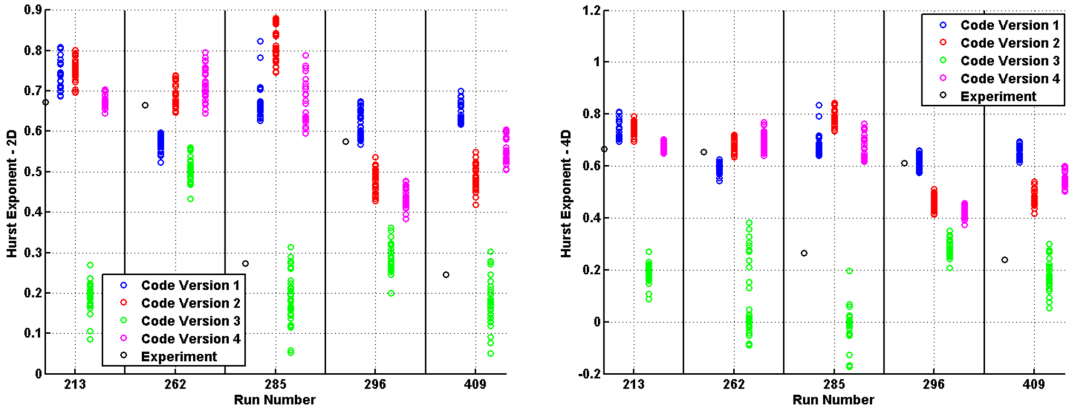


Figure 3.6: Calculated Hurst exponent values for each simulated trajectory, as well as for each experiment *left:* calculated values based upon 2-D trajectory information *right:* calculated values based upon 4-D trajectory information. Unlike the results for the correlation dimension there is substantial spread between individual simulations produced under the same conditions, although in many cases it is still possible to distinguish between values generated based upon different versions of the code.

Comparable results are shown for the Hurst exponent in Figure 3.6. Like the correlation dimension, there seem to be differences in the values calculated for the body linear, approximate-small, and nonlinear code versions. However, the differences are less immediately apparent due to increased variability among like populations. This may suggest a sensitivity of the metric to initial conditions, or alternatively suggest a higher convergence requirement. The mean value for the approximate-large version is significantly different from the mean values for the other code versions. However, this is attributed to the previously discussed tendency of approximate-large simulations to capsize. Taken together, these results indicate that the Hurst maybe somewhat-less useful than the perviously discussed correlation dimension for the validation of chaotic time series. Such a statement is indicative of a conservative viewpoint, as the results indicate a sufficient ability to distinguish between different code versions for certain individual runs, but not necessarily for all five experimental conditions. More

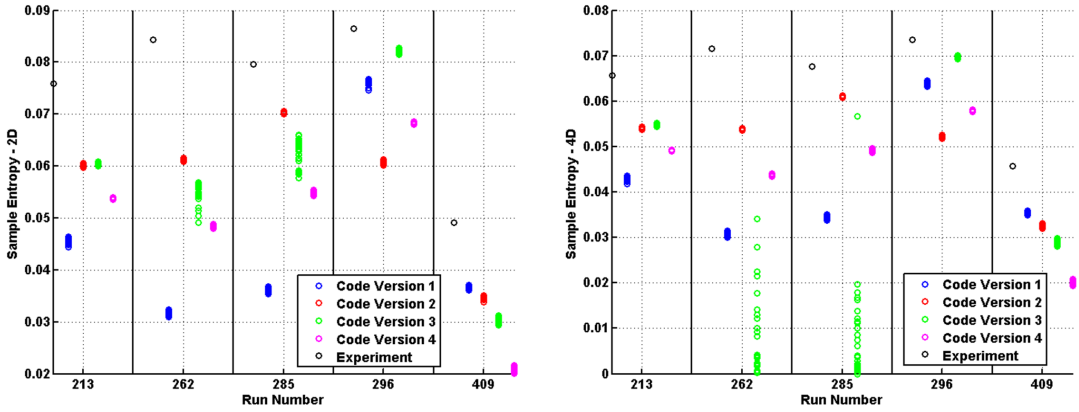


Figure 3.7: Calculated sample entropy values for each simulated trajectory, as well as for each experiment *left:* calculated values based upon 2-D trajectory information *right:* calculated values based upon 4-D trajectory information. Much like the correlation dimension, the observed behavior of this metric is highly desirable for a validation metric.

simulations may prove useful in determining the true usefulness of the Hurst exponent.

Figure 3.7 shows metric results for sample entropy. This was found to be well suited for use as a validation metric, with large changes in the mean metric values relative to the variance in the candidate validation metric values. Qualitatively, the results from the 2-D and 4-D data sets are very similar. The POD results are shown in Figure 3.8. For these results, trajectories were referenced to the mean fundamental mode of the nonlinear results for each wave condition (see the metric section in Chapter 2 for a detailed explanation of how POD metric values are calculated). Like the sample entropy, the POD results also show very high repeatability. However, POD also appears to be relatively insensitive to changes in the physics as well. This would indicate a metric of limited use for generic validation purposes, although it may work very well for certain cases with gross changes in attractor shape. This was observed to be the case with dramatic changes in attractor shape associated with bifurcations in Chapter 2. However, it does not appear to have the ability to resolve the more

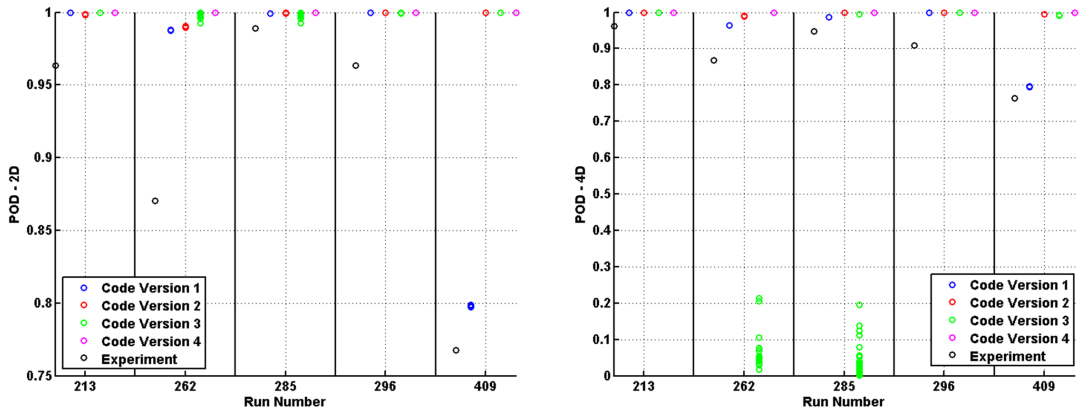


Figure 3.8: Calculated POD based metric values for each simulated trajectory, as well as for each experiment *left:* calculated values based upon 2-D trajectory information *right:* calculated values based upon 4-D trajectory information. This metric demonstrated limited utility as a validation metric for time series simulations.

subtle changes in dynamics between the code versions considered here.

Finally, the results for the Lyapunov exponent are shown in Figure 3.9. For the most part, the 2-D and 4-D results show similar variation in the metric values. Although, the 4-D results appear to be somewhat more sensitive to the changes in the physics than the 2-D results, this is slightly deceptive. For these five wave conditions, the code version 3 had a tendancy to capsize which affects the ability to find neighbors (which is how Lyapunov values are calculated here). Increasing the difficulty in finding neighbors did not affect the 2-D metric calculation as much as it did the 4-D metric calculations. After close inspection of Figure 3.9, it can be observed that there are comparatively few results that were able to be calculated for the code version 3 for the 4-D metric. Regardless, it can be concluded that Lyapunov exponents are a viable validation metric and should be considered further.

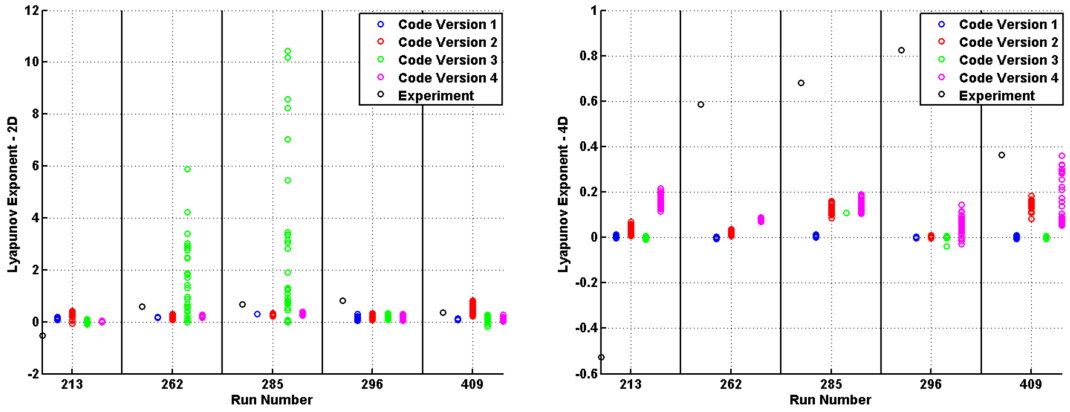


Figure 3.9: Calculated MLE values for each simulated trajectory, as well as for each experiment *left:* calculated values based upon 2-D trajectory information *right:* calculated values based upon 4-D trajectory information. The Lyapunov exponent demonstrated substantial potential as a validation metric, but there are practical concerns which arise in its implementation.

3.3.3 Metric values - large data set

3.3.3.1 Large set of simulations

Subsequent to the investigation of the smaller set of simulations, a larger set of simulations was performed, which included simulations matching 192 distinct experimental conditions. The total number of simulations for this set was 34,560 (192 conditions x 3 code variants x 60 initial conditions). This set of simulations was similar to the smaller set, but there were several important differences. For improved statistics, this set of simulations included 60 separate simulations for each combination of wave conditions and code version. Additionally, the range of initial conditions for roll and pitch velocities were increased to ± 10 degrees per second (the range of initial roll and pitch angles was kept at ± 10 degrees). Due to the increased number of experimental conditions and initial conditions included in this set of simulations, it was necessary to leave the nonlinear code version out of this set of simulations, due to its computational costs. This set of simulations included trajectories generated using the "linear," "approximate-small," and "approximate-large" versions of the ship motion

simulation code.

As discussed previously in the introduction, simulations were performed for 192 separate experiments using the linear, approximate-small and approximate-large code versions. Only the nonlinear code version was limited to the subset of five experiments. To gain some statistical insight into the validity of these candidate metrics, one way ANOVA was used to quantify how much of the total variance for an individual experiment can be attributed to differences in simulation physics (in other words, meaningful differences in the observed dynamics), as opposed to random errors or changes associated with the initial condition. Essentially, this represents a quantitative measurement of the characteristics qualitatively observed in the previous section for a smaller set of simulations.

For each validation metric, each individual wave conditions was classified as either "insignificant," "moderately significant," and "significant". The former category includes wave conditions for which the ANOVA analysis did not find a statistically significant difference between the metric values produced using the three code versions ($p>0.1$). The "moderately signifcant" category includes wave conditions for which the ANOVA analysis produce a p-value of ($0.05<p<0.1$). Finally, the "significant" category includes all wave conditions for which ANOVA produced a p-value less than 0.05.

The fraction of the total number of wave conditions falling into each category is shown in Figure 3.10 for 2-D metric calculations, and Figure 3.11 for 4-D metric calculations. In these figures, results considered "significant" and "moderately significant" correspond to cases where the candidate validation metric values calculated for each code version did not agree – in other words, the different physics models do not produce

the same time series dynamics. The majority of the simulations do fall in this category, although all metrics agree that differences are insignificant for at least a small minority of the experimental conditions. This minority may correspond to conditions where the dynamics are predominately linear, so that each code version captures the same dynamics.

For the most part, the results were found to be similar for both the 2-D and 4-D calculations, although there were minor difference. This similarity is in agreement with the observations from the small subset discussed previously. As a general rule, these results indicate that the 4-D calculation is somewhat more sensitive than the 2-D calculation, which is not surprising. Similar results calculated for combinations of only two of the calculation types (linear and approximate-small, for example) are shown in Appendix B. Examination of these results gives further credence to the results obtained using the full set of data. However, the paired results also demonstrate an important additional point, which is that different metrics are more sensitive than others to different changes in physics. There is value in testing with a suite of validation metrics, rather than a single, individual metric. Appendix C contains results for the POD metric, which was discussed in this chapter for the small set of runs but not the large set of runs. This is due to the fact that POD seemed to be signifcantly outperformed by the other metrics, particularly for the smaller data set. However, from the results show in Appendix C initial thoughts that this metric would prove useful for identifying gross physical changes in the system is correct. Additionally, the second and third modes were evaluated and those might prove to be helpful in identifying the more subtle changes occurring the in system.

The results shown here indicate that all of the metrics represent useful validation metrics. However, the results also suggest that the Hurst exponent may be distin-

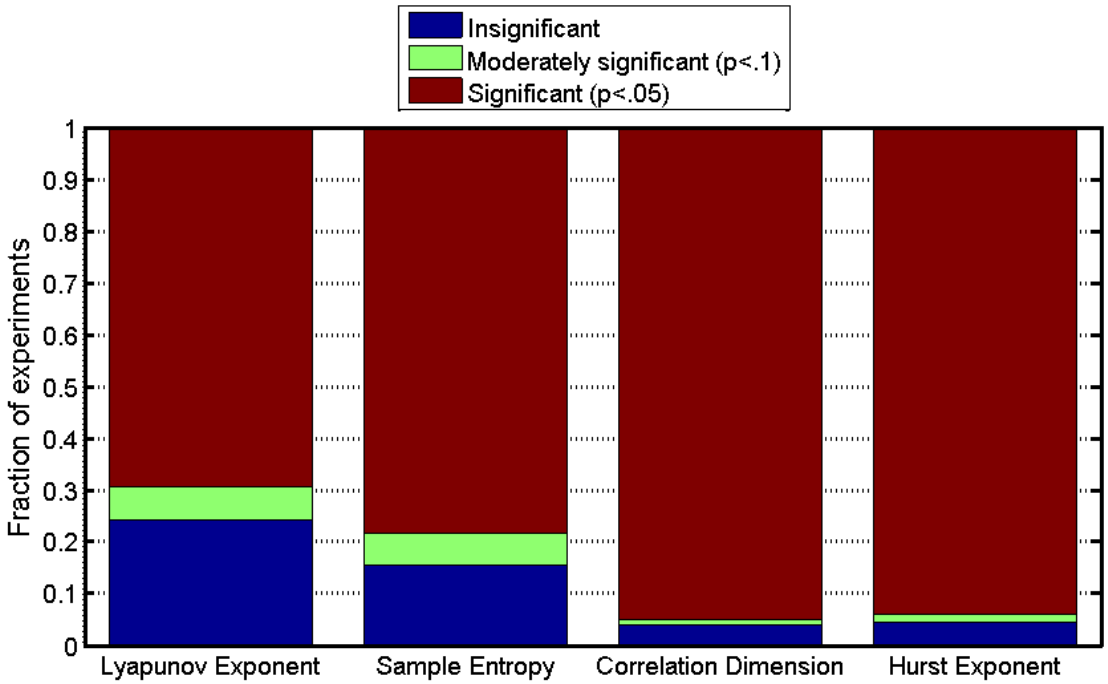


Figure 3.10: Classification of the statistical significance associated with metric values calculated using 2-D data for each set of wave conditions.

guishing between slight differences in dynamics not observed via the other candidate metrics; additionally, the Lyapunov exponent may be missing differences in dynamics that the other metrics observe. In the case of the Lyapunov exponent, this is likely due to the convergence issues associated with the numerical calculation of Lyapunov exponents directly from time series data (as opposed to an extended system, as was the case in Chapter 2).

3.3.4 Discussion

In the results for the small and large set of simulations, all of the metrics except for POD were found to have notable utility as metrics for the validation of time series simulations. Although it bears mentioning that from the results shown in Appendix C, POD may prove to be a quite useful metric (in the presence of large physical changes

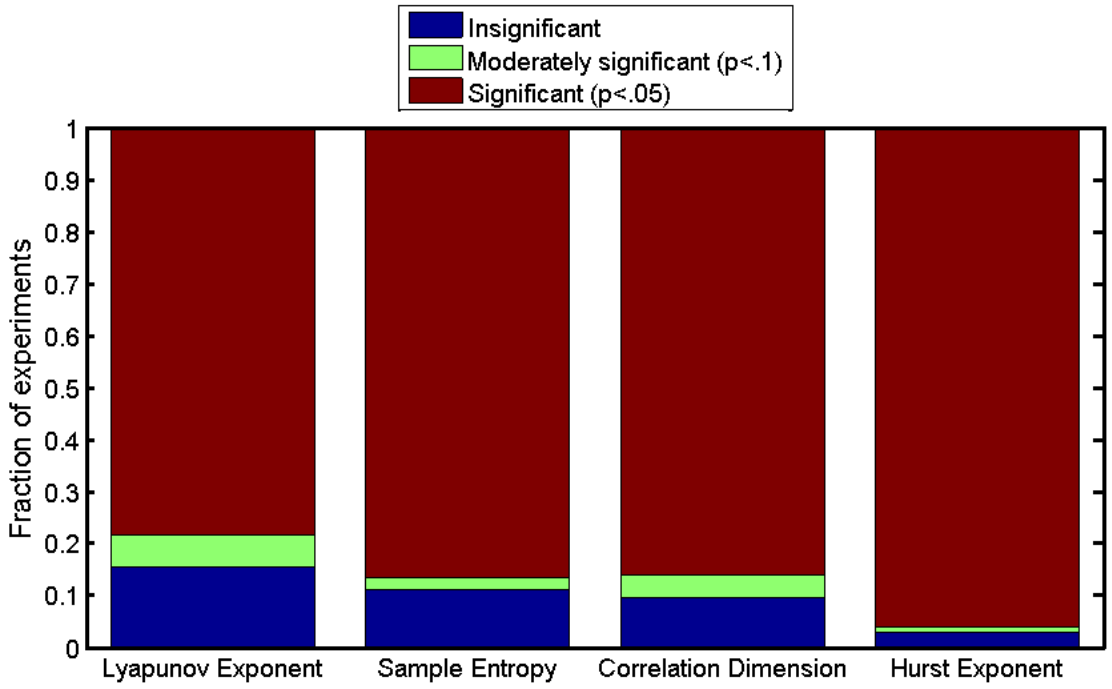


Figure 3.11: Classification of the statistical significance associated with metric values calculated using 4-D data for each set of wave conditions.

in the system) due to its ability to compare physics at multiple levels of sensitivity. This is particulary true of the correlation dimension and sample entropy metrics, which show near ideal behavior, with exceedingly low sensitivity to the inital condition and significant changes in response to different simulation physics. The Hurst and Lyapunov exponents showed similar changes in response due to changes in the model physics, but were demonstrably less repeatable due to increased sensitivity to the initial condition. However, statistical analysis found the changes in Hurst exponent associated with difference code versions to be very large in comparison to the spread in the data for each individual code version, suggesting that the behavior of the Hurst exponent is more ideal than one might expect based upon visual inspection of the results for the smaller set of simulations.

In many cases, it is may be either impractical or even impossible to measure the full state of the system. Under these circumstances, the ability of a validation metric to function based upon partial information is of critical importance. Since no one metric is perfect, each have their limitations (e.g. Lyapunov exponent are more reliably calculated with longer time histories as it is easier to find neighbors) this leads back to the previously suggested idea that a suite of validation tools would be the ideal way to validate a chaotic time series simulation. With this suggestion in mind and based upon the results found in this chapter, all metrics will serve as quality validation metrics to be included in a validation tool suite.

An additional factor which should be considered is the accuracy with which metric values can be evaluated for the experimental data set. In many cases, the acquisition of the experimental data sets are time consuming and/or expensive, resulting in the acquisition of a limited number of data sets. This lends additional importance to the insensitivity of the chosen validation metric to changes in the initial condition, as the

resulting variance associated with the initial condition may prove daunting to quantify. This problem is compounded under circumstances where error of the calculated metric values decreases at a rate above and beyond that provided by averaging the results from additional data sets. A principal example of such a metric in the present work is the Lyapunov metric, as discrete implementations of the Lyapunov exponent rely upon the presence of nearby, neighboring points which are not part of the local trajectory.

It is important to give consideration to the level of uncertainty present in the values of validation metrics. Proper model validation compares a statistical distribution for the validation metric value with a corresponding distribution for the experiment (Oberkampf and Roy, 2010); such distributions are generally obtained via some form of uncertainty analysis. For simulations this may involve quantification of the various error sources affecting the final result (truncation error, convergence errors, etc.). Correspondingly, an uncertainty analysis should be performed for the experiment; this usually involves quantifying the errors of various sensors and measurements. A rigorous comparison of the simulated and experimental metric values is not performed here due to a lack of such information for the experiment. Generally, the decision regarding the validity of the model reduces to the probability that the metric values calculated for the simulation and the experiment came from the same distribution. In the cases where very high uncertainties are present in the calculated validation metric values, such a conclusion may be foregone, regardless of the actual similarity in dynamics between the simulation and the experiment. As a result, validation metrics should generally be selected which will have small uncertainties associated with their calculation (both for the simulation and for the experiment).

3.4 Conclusions

In this work, a focused investigation into the validity of five different measures of nonlinear system behavior has been undertaken. This investigation built upon the results described in Chapter 2, by investigating the use of these metrics to validate a complex ship dynamics model, as opposed to the behavior of canonical dynamical systems. Based upon the results presented in this chapter (both the large and small sets of simulations), it was concluded that out of the five metrics considered, the correlation dimension and sample entropy have the most utility for the validation of (chaotic) time series simulations. In the small set of simulations, the Hurst and Lyapunov exponents also demonstrated significant sensitivity to changes in physics, but had appeared to have more variability (as the initial condition changed, and not associated with a change in physics) compared to the correlation dimensions and sample entropy. The results from the large set of simulations lended validity to this conclusions for the Lyapunov exponent, but suggested that changes in the Hurst exponent were actually more statistically significant than those of any other metric. Ultimately, all four of these metrics represent good choices for the validation of chaotic time series simulations. Upon further investigation, the POD based metric, while potentially not as sensitive as other metrics, has hinted at being able to detect more subtle physics when looking at the higher order modes. Additionally, POD converges quickly and would be very helpful when one is dealing with shorter time histories.

There are several possible avenues of future work which might build upon this. It is obvious that there are many potential validation metrics other than those considered in this work; these represent only a small sampling of the most obvious candidates. There would be significant value in investigating other candidate metrics as well. Ultimately, an ideal scenario would involve a suite of well behaved, easily calculated metrics, so that validation decisions can be based upon the information from multi-

ple metrics. This would aid in overcoming the inherent difficulties associated with validating the simulation of chaotic times series phenomena.

CHAPTER IV

Experimental Study on Deformation of Flexible Wedge Upon Water Entry

4.1 Introduction

The proper choice of validation metric is critical to any effort to accurately assess the agreement between a numerical model and physical reality. However, the setup and design of the experiments providing data for comparison is at least as important, if not more so. In this study, physical experiments were performed to provide data for the validation of numerical models of the water impact of complaint ship structures, such as hovercraft.

The impact of ships in the water (regardless of whether the structure is compliant or rigid) is known as slamming. Slamming can be described as the hull lifting out of and then re-entering the water at some greater than critical re-entry velocity, and represents a highly nonlinear ship dynamics phenomena. For a detailed review on the definition and nature of different slam events, the interested reader is referred to

This chapter is derived in large part from Cooper and McCue, 2011.

Riley et al (2010). One important slamming scenario is that of a compliant structure (this is very relevant to hovercraft). Sophisticated models have been developed to describe the dynamics of this scenario, but there is a a lack of experimental data for the slamming of compliant structures to use validating such models.

Slamming results in very large loads being applied to ship structures. These loads represent an extremely important factor that must be accounted for during the design of ships, since they represent one of the most severe loadings experienced. As a result, understanding the mechanics of such events is very important in the ship dynamics community. Much previous work has been performed studying this phenomenon for rigid structures. This body of work includes theoretical (Cooker and Peregrine 1995; Carcaterra and Ciappi 2004; Korobkin 2004; Korobkin and Iafrati 2005; Korobkin 2007) and computational studies (Gaudet et al. 1998; Battistin and Iafrati 2003; Battistin and Iafrati 2004; Oger et al. 2006; Federico and Amoruso 2009), as well as experimental. Much of the experimental work to date has focused on the hydrodynamic aspects of such water entry, as opposed to the structural deformations, which is reasonable when considering mostly rigid structures. Such studies include the work of De Backer et al. (De Backer et al. 2009), who studied the impact of axisymmetric spheres and cones, Judge et al. (Judge and Perlin 2004), who studied the asymmetrical impact of cylinders, and Yettou et al. (Yettou et al. 2006), who studied the impact of wedge geometries. In these studies, the pressures applied to the rigid geometries are a primary interest, along with the cavity formation that occurs during water entry.

The study by Okada et al. (Okada and Sumo 2000) measured the strains that occurred in aluminum plates during water entry, along with the previously described pressures and cavities. However, to the knowledge of the authors, such studies examining the deformation of structures during impact have not been performed for more flexible

membrane-like structures. Such structures are important, as they are analogous to those of air cushion vehicles (ACV) and surface effect ships (SES). Moreover, it is clear that there are several important differences in the dynamics of the water entry of such structures and the dynamics of more rigid structures. For example, water impact of compliant structures can reduce the damping of the cobblestone oscillations generated by wave encounter (Faltinsen 2000). This is due to the lack of an air gap being generated between the surface and water. While such gaps are normally produced in the impact of rigid structures, they are generally either absent or greatly reduced in the impact of very compliant structures (Ulstein and Faltinsen 1994; Ulstein and Faltinsen 1998). Additionally, the restoring force in such cases is very different than that in hydroelastic studies performed with fairly rigid structures (such as the aforementioned study by Okada et al.). In the case of compliant structures such as those considered in the current study, restoring forces are almost entirely due to membrane tensions and/or internal pressure within the structure, as opposed to the mechanical properties of the structural material. More recently, Panciroli (2013) described multiple unique scenarios of fluid-structure interaction which can occur during the impact of deformable wedges. Panciroli et al. (2013) also discovered that the specific type of behavior observed heavily depends upon the ratio of the natural frequency fo the structure to the characteristic wetting time.

The goal of the current study is to examine the deformation of such compliant structures during water impact so as to provide experimental data for the validation of the previously mentioned models of the slamming of compliant structures. The geometry considered in these experiments is that of a wedge and consisted of a silicone rubber membrane wrapped around an aluminum frame to form the wedge shape, with several different tensions considered. A high speed camera was used to record the positions of dots placed on the rubber membrane, allowing calculation of the strain of the mem-

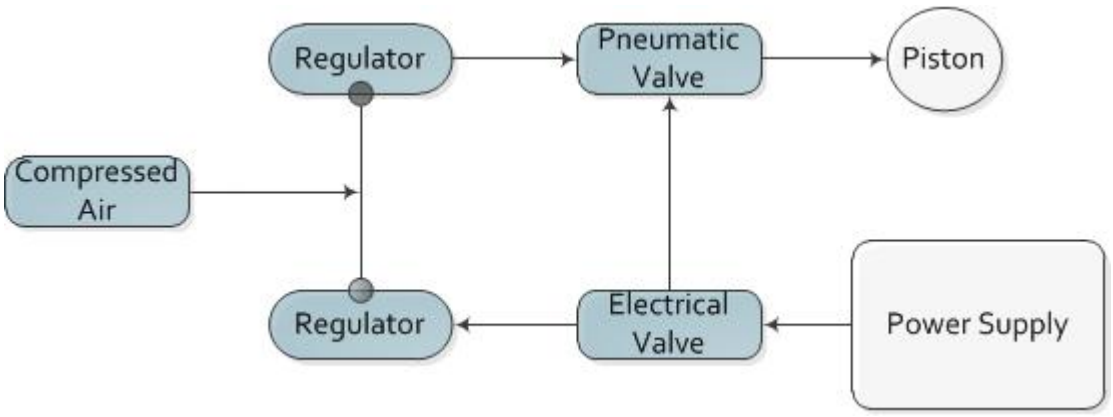


Figure 4.1: Schematic of experimental setup including all necessary parts required to fire the pneumatic piston.

brane during impact. The results of this study show that the membrane undergoes substantial deformation during such impact even at high tension, with the membrane vibrating at frequencies determined primarily by the tension of the membrane.

4.2 Experimental Setup

The experimental setup can be divided into three main components, the driving apparatus, the wedge, and the optics.

4.2.1 The Driving Apparatus

A source of compressed air, an air tank in this case, section of plumbing for air transport to the piston, as well as regulators, valves and a power supply were needed to fire the piston and drive the wedge into the tank of water. Figure 4.1 shows a schematic of these components and how they were connected.

The piston was mounted on a "3/4" aluminum plate supported by double slotted 80-20 secured to an optical table. A tank (2' by 1' by 1.5') was filled 3/4 of the way full and was centered beneath the piston. The tank was also secured to the optical

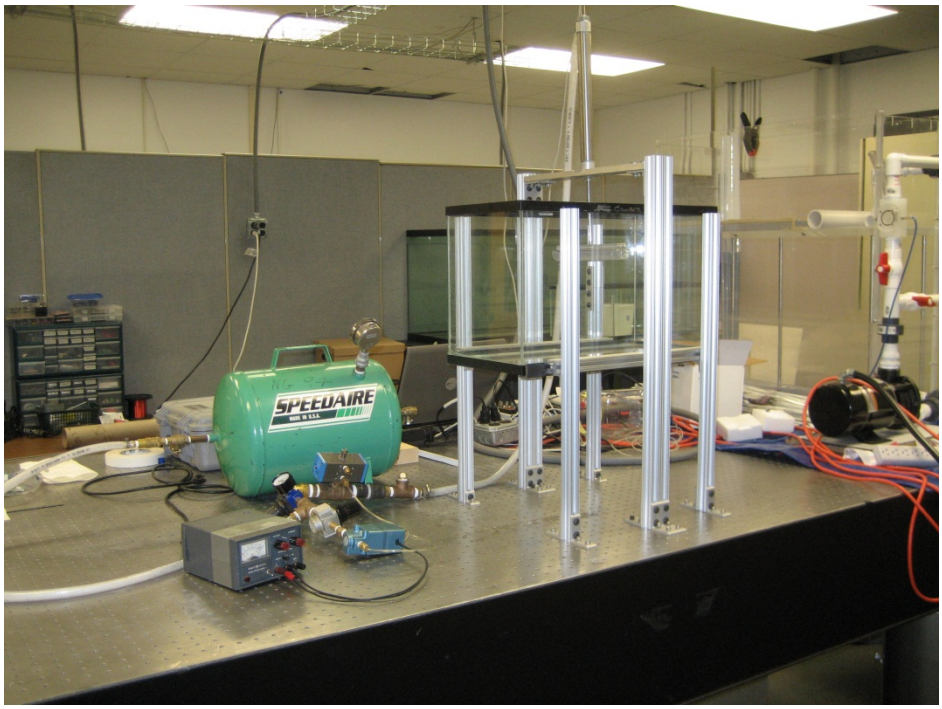


Figure 4.2: Experimental setup. A pneumatic piston mounted above the tank was used to drive the assembled wedge into still water in the tank, as shown in the image. Compressed air was provided from a tank; this was refilled as necessary. A pressure regulator was used to control the pressure supplied to the piston, while a solenoid valve was used to control the flow of air, allowing the wedge to be driven into the water on command.

table using 80-20 pieces to avoid jostling during experimentation. An image of the assembled experimental setup is given in Figure 4.2.

4.2.2 Wedge

The wedge was milled from an aluminum plate and has two separate parts. The first piece is the lower t-bar, which is 6 inches long, 1 inches wide, and of an inch tall. The top portion consists of a flat plat of corresponding length and width. The two pieces can be connected together via carriage bolts and serve as a clamp for a silicone rubber membrane that is wrapped around the t-bar frame. Figure 4.3 is an image of the frame that supports the silicone rubber membrane that completes the wedge. The rubber membrane stretched over the wedge consisted of a .35 mm thick layer

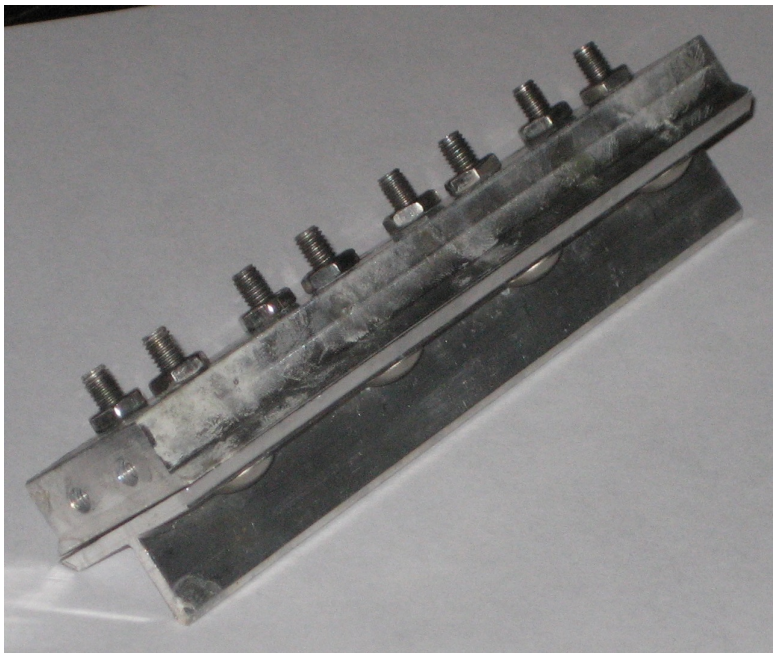


Figure 4.3: Metal frame of the wedge that supports the structure of the silicone rubber membrane.

of silicone rubber. A grid of dots was added to the surface of the rubber membrane wrapped around the wedge. These dots were used to measure the strain present in the rubber when it is initially stretched, which along with knowledge of the mechanical properties of the rubber material, allowed calculating of the tension. Additionally, these dots allowed calculation of the transient strains occurring in the membrane during water entry, as their relative deformation could be tracked using imagery from a high speed camera. In low tension cases, these dots were placed on the rubber prior to stressing the membrane, in the highest tension case, it was necessary to mark the dots after stressing the membrane. In this case, a measurement of the positions of the dots was taken in the relaxed position (after testing) to allow for calculating of membrane tension, which is justified due to the materials elastic nature. Figure 4.4 shows an example of the dots placed on the membrane rubber.

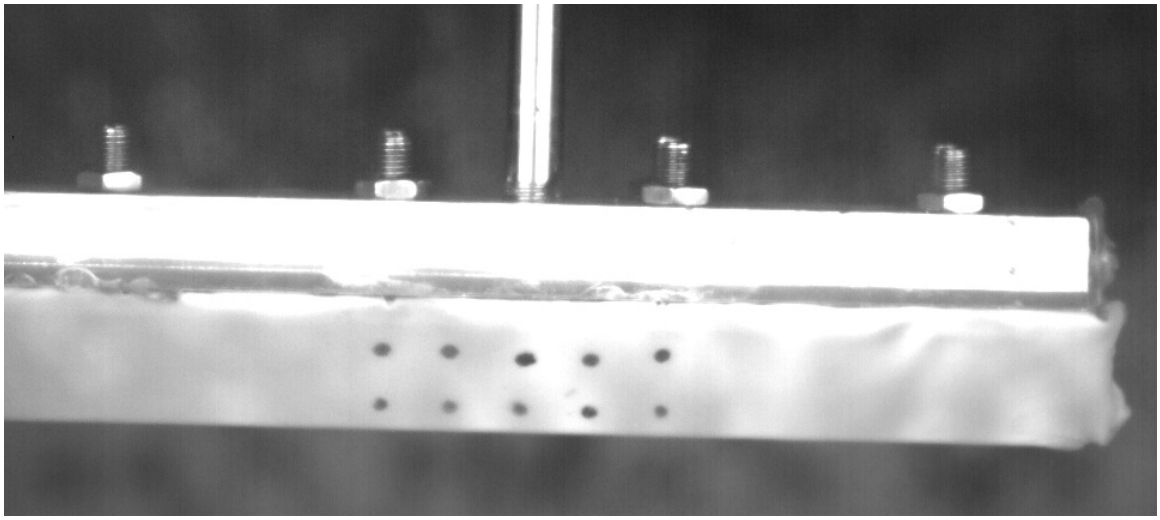


Figure 4.4: Example Image showing the placement of dots on the rubber membrane of the wedge. These dots allowed for calculation of both tension as well as transient strains during water entry.

4.2.3 Rubber Properties

Since knowledge of the mechanical properties of the rubber was necessary to allow for calculation of the tension present in the membrane, mechanical testing was performed to determine the elastic modulus of the rubber. This was performed under quasi-steady state conditions, with the applied load being slowly ramped upward. Dogbone specimens (Qualitest, Standard # D 638-5-IMP, Serial # 9-1-1) were tested, using an Instron machine (Instron 5867). A representative stress-strain curve for the membrane material is shown in Figure 4.5. It is apparent from this figure that the behavior of the material is extremely linear outside of an initial toe region observed early in the loading curve. From this data, an elastic modulus of approximately 75 kPa was calculated. The fundamental resonant frequency of the stretched membrane can be calculated from this tension according to Equation 4.1:

$$f = \frac{1}{2w}\sqrt{\frac{T}{\rho_a}} \tag{4.1}$$

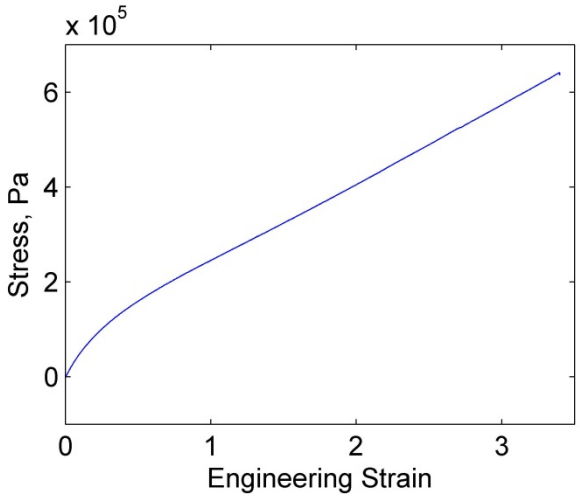


Figure 4.5: Stress-strain curve obtained from quasi-steady tension testing of dogbone specimens of the rubber used to form the outside walls of the wedge. The behavior of the material is mostly linear, with a small initial toe region observed. The elastic modulus of the material is approximately 75 kPa.

where T represents the tension in the membrane, ρ_a represents the area density of the membrane, and w represents the width of the membrane (the vertical dimension in Figure 4.4). Note that Equation 4.1 is the same as the equation for a string, except for the substitution of area density for linear density. This is valid since both ends of the membrane are essentially free, leading to uniform deformation of the membrane along the length of the wedge.

4.2.4 Optics

Video of the wedge impact was recorded using an IDT X-5 camera. The frame rate for all tests was 1 kHz, while the exposure time was set differently for each driving pressure, in order to avoid blurring of the wedge images. Two large lights were used to illuminate the wedge for imaging. Image acquisition was started prior to opening the valve to drive the wedge into the water, with four thousand frames, spanning 4 seconds, being recorded. This time was sufficient to ensure that the entire entry was recorded.

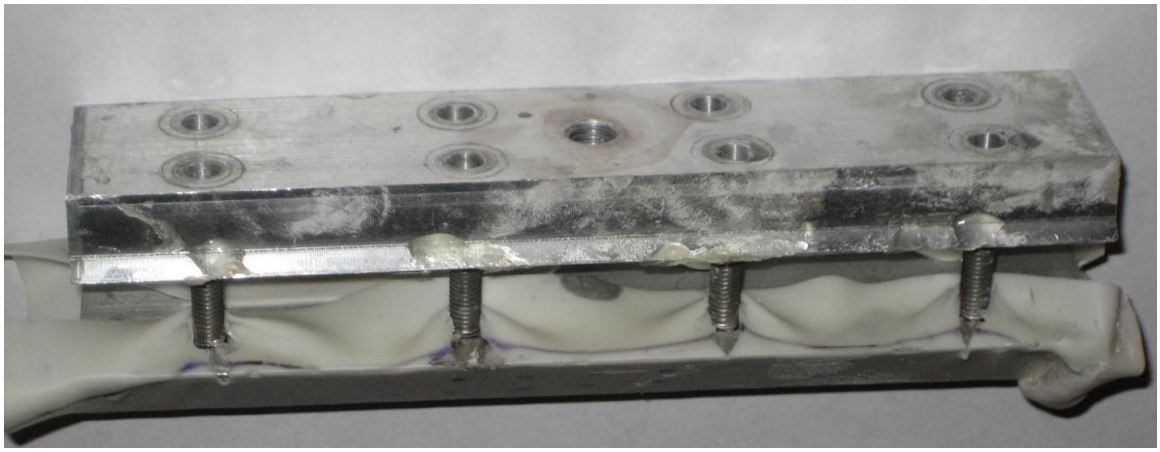


Figure 4.6: Construction of the wedge. Posts on the top of the wedge were placed through holes in both sides of the rubber membrane. The top of the wedge was then bolted down on top of the posts, locking the membrane into place.

4.2.5 Test Matrix

Two different membrane tensions were considered in this study. These tensions were controlled by cutting the rubber to specified lengths prior to wrapping it over the wedge. Posts on the top of the wedge were used to hold the rubber in place prior to clamping down on it, so the locations of the holes for the posts were the controlled length, not the overall length of the membrane. These aspects of the wedge design are shown in Figure 6. Three different driving pressures were supplied to the pneumatic piston; these pressures were 10, 20, and 30 psi. 40 psi was attempted, but the wedge was moving too fast to be effectively imaged and analyzed, as images were blurry and comparatively few frames could be obtained before the piston reached the end of its travel. The test matrix is shown explicitly in Table 4.1.

Pressure (psi)	Tension 1 (10 N/m)		Tension 2 (235 N/m)	
	Entry Velocity (m/s)	Resonant Frequency (Hz)	Entry Velocity (m/s)	Resonant Frequency (Hz)
10	34	24	34	115
20	68	24	68	115
30	98	24	98	115

Table 4.1: Test matrix for the current study. Two different membrane tensions were considered, along with three different driving pressures applied to the piston. The resonant frequency for each tension and the entry velocity of the wedge corresponding to each driving pressure is also given.

4.3 Data Processing

Data processing was performed using the freely available particle tracking code (GNU license) provided by the Blair and Dufresne, available online (Blair and Durfesne, 2005). This code is a MATLAB implementation of an IDL particle tracking software. As a result, the data processing in this study begins by first inverting the image so that the grid of points is bright and the background is dark. The next step is to crop the entry cavity out of the image and pass this image through a spatial band-pass filter. The pass wavelength of this membrane was set to be equal to the apparent size of the dots marked on the rubber membrane. Peak detection and sub-pixel interpolation to determine the centroids of the peaks was performed using functions provided along with the aforementioned particle tracking algorithm. In each step, particle positions were retained only if the number of detected particle positions was less than some threshold (usually set to fourteen). The purpose of this was to avoid overloading the particle tracking algorithm.

Once the peaks were detected for each frame, the particle tracking algorithm was called to perform the combinatorics (equivalent mathematically to the traveling sales-

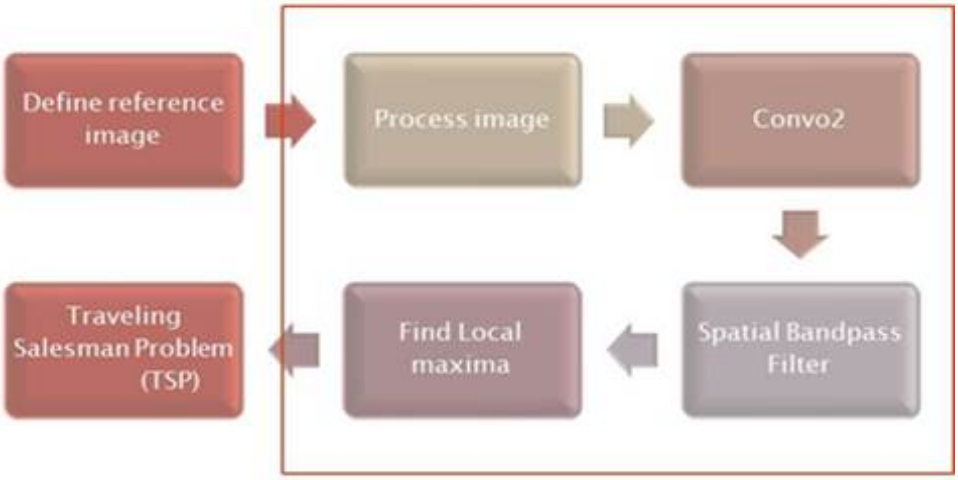


Figure 4.7: Flowchart of the data processing steps roughly detailing the major steps in processing the images collected by the high speed camera. "Convo2" refers to a 2-D image pair convolution.

man problem) necessary to assemble particle trajectories. Once the trajectories were assembled from the recorded particle positions, an estimate of the strain in the membrane was obtained from each vertical pair of dots on the wedge membrane, with the strain in the membrane at any given time taken to be the average of all valid strain measurements obtained at that time. An overview of the data processing algorithm used in this study is shown in Figure 4.7.

4.4 Results and Discussion

Before considering the strain measurements obtained in this study, it is worthwhile to note that there was no observation of the development of an air gap between the wedge membrane and the water throughout this experiment. The development of such a gap would have been fairly obvious in the experiment, as it would have distorted the imaging of the wedge during water entry (due to the index of refraction mismatch between air and water), making strain measurement difficult if not impossible. This observation is in agreement with prior works (Faltinsen 2000), but still

bears mentioning.

The general trend observed in the experimental strain data is one of decaying sinusoidal strains in the wedge membrane after water entry. Figure 4.8 shows the median time histories for each set of test conditions (membrane tension and driving pressure). Here the solid lines represent the median strains, while dashed lines indicate plus or minus one standard deviation. Note that the median is presented (as opposed to the average) as an alternative to applying outlier detection to the data. Wake and cavity effects tended to block observation of the wedge immediately following water entry. At the end of the run, when the piston reached the end of its travel, observation was also difficult, due to wake effects as well as large deformation of the wedge. As a result, transients appear in the data at the beginning and end of each run; these represent misidentification of points due to observational difficulties, rather than any real phenomenon.

Figure 4.9 shows the (absolute value) of the frequency spectrum of the median strain curves shown in Figure 4.8. A Hamming window was used to remove the influence of end effects on the frequency spectra (Roberts 2004). Here, the blue line represents the absolute value of the frequency spectra, while the vertical black line represents the resonance frequency calculated for the appropriate membrane tension. Studying these results, we observe that in each case there is a strong spectral content at a frequency near the estimated resonant frequency of the membrane. From this observation, we can conclude that the observed vibrations in the wedge membrane are most likely due to free vibrations of the structure, being excited by the initial impact of the wedge with water.

The vibration of the wedge appears dominated by the structural properties of the

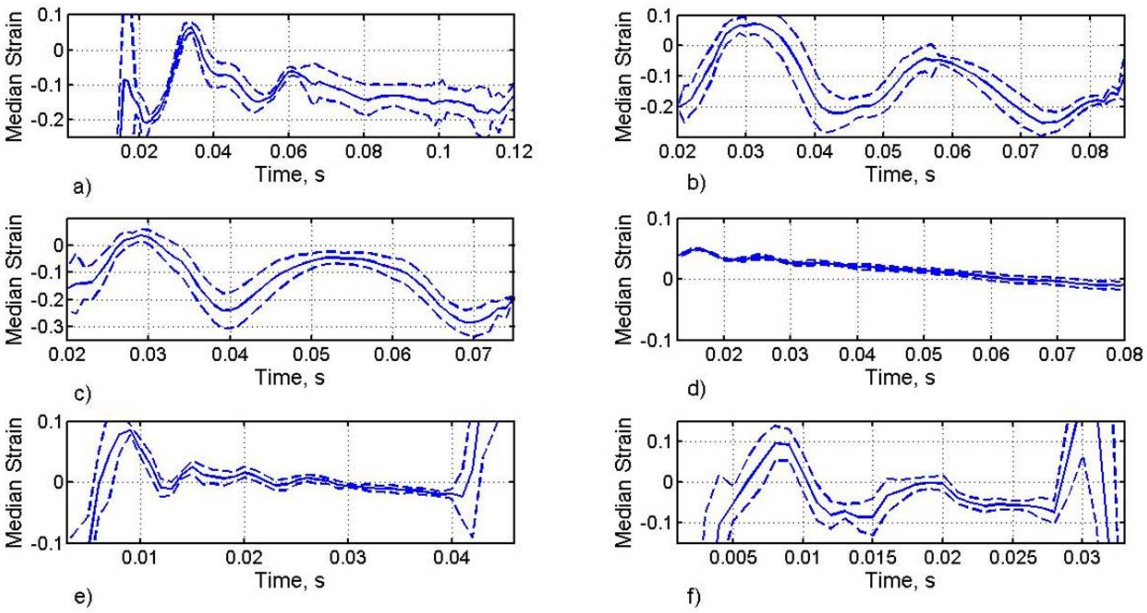


Figure 4.8: Median strain histories for each test case. In this data, time is measured from water entry of the wedge. The dark blue line in each case represents the median strain over all runs, while the dashed lines represents plus or minus one standard deviation from the median strain. It should be noted that large strains at the beginning and end of runs are due to misidentification of dots in the post-processing, and do not represent any physical phenomena. a) Driving pressure of 10 psi, membrane tension of 10 N/m. b) Driving pressure of 20 psi, membrane tension of 10 N/m. c) Driving pressure of 30 psi, membrane tension of 10 N/m. d) Driving pressure of 10 psi, membrane tension of 235 N/m. e) Driving pressure of 20 psi, membrane tension of 235 N/m. f) Driving pressure of 30 psi, membrane tension of 235 N/m.

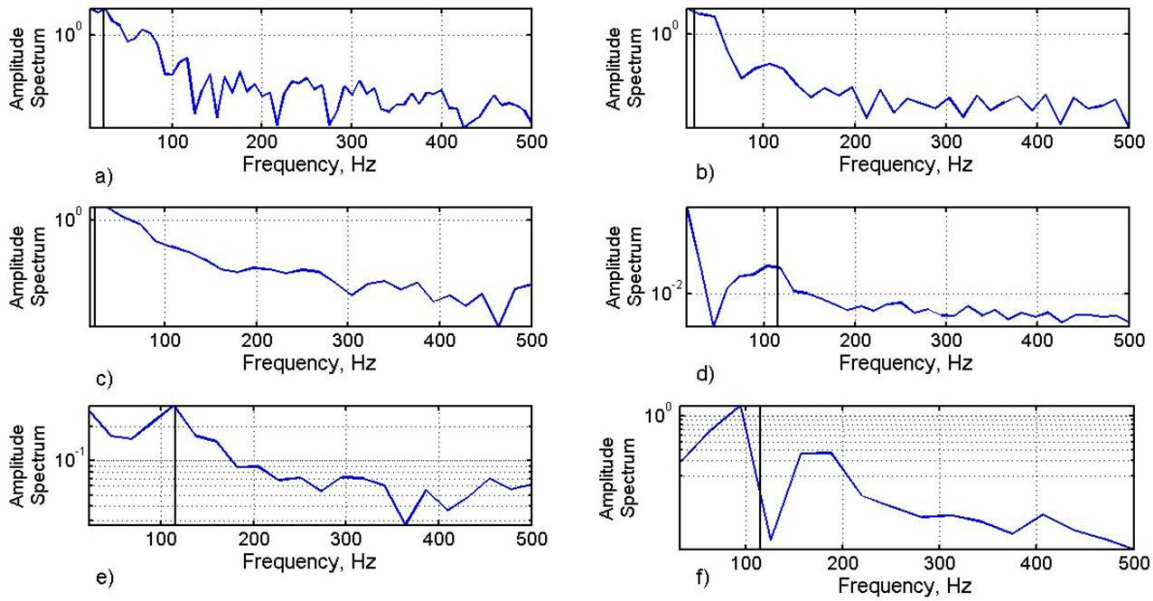


Figure 4.9: Absolute value frequency spectra of the median strain histories for each case shown in Table 4.1. Vertical black lines represent the resonant frequency of the wedge membrane in that test case. Good agreement was observed between increased strain spectral content and the resonant frequency of the membrane. a) Driving pressure of 10 psi, membrane tension of 10 N/m. b) Driving pressure of 20 psi, membrane tension of 10 N/m. c) Driving pressure of 30 psi, membrane tension of 10 N/m. d) Driving pressure of 10 psi, membrane tension of 235 N/m. e) Driving pressure of 20 psi, membrane tension of 235 N/m. f) Driving pressure of 30 psi, membrane tension of 235 N/m.

membrane, with the initial impact with the water providing an excitation of the fundamental mode of the membrane. It is likely that other modes are also excited (such as is the case when a bar is struck by a hammer), however, the frame rate of the camera is insufficient to resolve higher modes.

Several avenues for future extensions of the current work exist. In the present study, it would have been ideal to have placed strain gauges on the membrane, however, the compliant nature made this impractical for the study at hand. Repeating the experiments with a more accurate form of strain measurement would be an important extension of this work, as it would allow for firmer conclusions to be drawn. An additional limitation of the current work was that the 1 kHz frame rate of the cameras used in this study was insufficient for measuring higher order modes of the membrane, particularly in the high tension case. Since the frame rate of the cameras used in this study is already quite high, higher frame rates may not be practical. This again indicates the usefulness of repeating the experiment with non-optical strain measurements, although this would be difficult at best. Along similar lines, direct measurements of pressures along the membrane would be of great use, though installing pressure taps in the flexible membrane without altering the flow would prove to be very difficult.

Another goal of this work is to provide a qualitative and quantitative data set for the validation of computational simulations of water entry. By performing these experiments with an idealized geometry that is easily observable, equivalent simulations could be performed, allowing validation of a code for water entry simulations. Afterwards, the validated code could be applied to other geometries which are much less accessible experimentally. This was the primary motivation for the current study, and represents an important application of this work.

4.5 Conclusions

In this work, the water entry of a compliant wedge shaped membrane was studied experimentally. Strains during water entry were recorded using a series of dots placed on the membrane, along with a high speed camera used to image these dots during water entry. Froude numbers for the experiment were in the vicinity of 1, while Reynolds and Weber numbers were on the order of 10,000 and 100, respectfully. Examination of the spectral content of the strains recorded in this study found good agreement between the predicted resonant frequency of the membrane and increased spectral content in the strain measurements. It is believed that water entry supplied an initial impulse to the membrane, which henceforth oscillated almost entirely as a free membrane, with the water also acting to increase the damping of oscillations.

The strain measurements obtained in this study were limited in some aspects by the optical method used to measure them. Most limiting was the inability of the technique to resolve the strains at the time of water entry, as the membrane was blocked from optical access by the waves and cavity being generated by entry. Ideally, future experimental studies building upon the current study would overcome this problem using non-optical measurements; however, obtaining non-optical strain measurements in such compliant materials is difficult at best. Additionally, it is hoped that the measurements obtained in this study will be of use for the validation of numerical algorithms for simulating the water entry of such structures.

CHAPTER V

Non-intrusive Polynomial Chaos for Efficient Uncertainty Analysis in Parametric Roll Simulations

5.1 Introduction

There exist many applications where a validated model has either random or unknown inputs and as a result must be treated as a random process when applied to real world applications. One such application worthy of particular note is uncertainty analysis, although there are certainly many others. Such applications compound the computational costs associated with the model, as they inevitably involve a large increase in the total number of simulations performed. The most common method of characterizing the statistical distribution resulting from such a random process is the Monte Carlo method, which relies on evaluating the random process for a large number of random inputs. The Monte Carlo method was first formally introduced by Stanislaw Ulam in 1946 following the Manhattan Project. During the Manhattan Project, it was first applied to the study of neutron diffusion by Ulan and John van

This chapter is derived in large part from Cooper and McCue, 2015.

Neumann (Metropolis, 1987).

Although the Monte Carlo method is quite accurate, it requires a very large number of simulation runs and can be extremely computationally expensive. For numerical models which are very computationally expensive, the Monte Carlo method may well be unfeasible. One alternative to the Monte Carlo method is the polynomial chaos method, in which the dependence of random process upon its inputs is approximated by an orthogonal polynomial expansion (Hosder, 2006). This approach has the potential to greatly reduce the number of simulations necessary to accurately predict the distribution of a random process. Previously, this method was applied to the Mathieu equation and a 1.5 degree of freedom parametric roll model in regular seas, which was able to accurately describe the Gaussian type distribution of these random processes with several hundred-fold decreases in computational time (Wu et al, 2011). However, this method comes with some potentially problematic issues, namely that it may not provide accurate results for simulations that demonstrate sudden changes in behavior, such as bifurcations.

The purpose of this chapter is to further explore this alternate method of evaluating random processes using a multi-degree of freedom ship motion code to model the maximum roll angles experienced by a vessel during parametric roll. As in the Wu et al (2011) study, this thesis utilizes a Gaussian distribution for the random variables, which does not preclude non-Gaussian results. The numerical simulation tool used represents a much more complex random process than those studied by Wu et al, and thus presents a more sophisticated test of the PC method.

5.2 Polynomial Chaos

Homogenous chaos was first defined as the span of Hermite polynomial functionals of a Gaussian process in Weiner (1938). Polynomial Chaos is defined as a member of this set of functionals. Cameron and Martin (1947) expanded the application of Polynomial Chaos by proving that the Fourier-Hermite polynomial series converges to any L2 functional in the L2 sense, meaning that Hermite polynomial expansions can be used to describe any stochastic process. Initial uses of Polynomial Chaos were primarily focused in the solid mechanics and fluid mechanics fields. Ghanem and Spanos (1991), and later Ghanem (1999), made contributions in solid mechanics by combining Polynomial Chaos with finite element methods to determine model uncertainty, while Xiu and Karniadakis (2003) applied Polynomial Chaos to modeling uncertainty in fluid mechanics. In addition, Xiu and Karniadakis (2002) provided a general approach to representing uncertainty using polynomial chaos.

In current practice, polynomial chaos is used to approximate a random process using the weighted sum of a set of orthogonal polynomials: (Hosder et al, 2006).

$$\alpha^*(t,\xi) = \sum_{P-1}^{i=0} \alpha_i(t)\Psi_i(\xi) \tag{5.1}$$

where α^* is the random process, $\alpha_i(t)$ is the deterministic part, and $\Psi_i(\xi)$ is the orthogonal basis function of the ith mode. Here, ξ is a n-dimensional random vector with a previously defined probability distribution. The total number of modes included in the approximation is given by:

$$P = \frac{(n+p)!}{n!p!} \tag{5.2}$$

where n is the number of random variables and p is the desired order of Polynomial

Chaos.

In Weiners (1938) original work Hermite polynomials were used as the orthogonal polynomials. More recently, other types of polynomials have been used to construct Polynomial Chaos expansions depending on the type of random variable (Xiu and Karniadakis, 2002). For example, Laguerre polynomials are used in processes involving gamma random variables and Charlier polynomials are used with Poisson random variables. However, it has been shown by Lucor et al (2001) that Hermite polynomials provide optimal convergence rate for Gaussian random processes. As a result, since the current study uses Gaussian inputs, Hermite polynomials are used throughout with the acknowledgement that there is potential for a non-Gaussian output.

One of the most useful properties of NIPC is that the statistical moments can be calculated directly from the basis polynomials and their coefficients. The mean of a random process is given by the 0th order polynomial coefficient, so that:

$$E[\alpha^*(t,\xi)] = \alpha_0(t) \tag{5.3}$$

Meanwhile, the variance of the random process can be calculated from the rest of the polynomial coefficients:

$$Var[\alpha^*(t,\xi)] = \sum_{i=1}^{P} \alpha_i(t)^2 \langle \Psi_i^2(\xi) \rangle \tag{5.4}$$

Note that these expressions are only applicable for the case where the input distributions have zero mean and unit variance. Formulating the problem so that this is the case is generally trivial an offset and a multiplier for each input variable can simply be incorporated into the random process.

5.3 Methods

Polynomial chaos can either be performed intrusively or non-intrusively. An intrusive application of polynomial chaos requires modification of the original computer codes, while the Non-Intrusive Polynomial Chaos (NIPC) method treats the original code as a black-box, which need only return the value of the random process for a given set of input values (Hosder et al, 2006). Thus, NIPC allows the random process to be represented as polynomial expansions based solely upon the evaluation of the random process for a small set of random inputs.

5.3.1 NIPC Collocation

One method used by Hosder et al to solve for the coefficients ai is the collocation method (Hosder et al, 2006). The collocation method involves expanding Equation 5.1 for some number of arbitrarily selected random vectors ξ_j, so as to obtain the linear system:

$$\begin{bmatrix} \Psi_0(\xi_0) & \Psi_1(\xi_0) & \dots & \Psi_P(\xi_0) \\ \Psi_0(\xi_1) & \Psi_1(\xi_1) & \dots & \Psi_P(\xi_1) \\ \vdots & \vdots & \ddots & \vdots \\ \Psi_0(\xi_m) & \Psi_1(\xi_m) & \dots & \Psi_P(\xi_m) \end{bmatrix} \begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \vdots \\ \alpha_P \end{bmatrix} = \begin{bmatrix} \alpha^*(\xi_0) \\ \alpha^*(\xi_1) \\ \vdots \\ \alpha^*(\xi_m) \end{bmatrix} \tag{5.5}$$

which can then be solved for the projection coefficients α_i. It should be noted that the above expression does not require that the number of arbitrarily selected random vectors be equal to P; if this is not true, valid solutions can still be obtained by solving the system in the least squares sense.

5.3.2 Response Surface Method (RSM)

The response surface method, which was proposed by Box and Wilson in 1951, is a commonly applied method which attempts to approximate a random process with a multivariable polynomial. In most applications this polynomial is of second order.

The coefficients of the fitted polynomial are chosen so as to match a small (compared to Monte Carlo analyses) population of input data points and their corresponding outputs as closely as possible. While the RSM method is similar to NIPC in that both involve the approximation of a (random) process with a multivariable polynomial, there are differences in how the coefficient values are determined. Unlike the NIPC collocation method (which also fits a polynomial to a series of points via least squared or similar methods) points for RSM method are generally organized into some type of grid, while randomly selected values are used with the collocation method. There are a wide variety of different gridding methods that have been proposed by different authors for use with the RSM method. In this case, RSM was applied using randomly generated samples, which were pulled at random from the corresponding Monte Carlo result sets. The most significant difference between RSM and the collocation approach to NIPC is the choice of polynomials used to model the random process. With RSM, monomials are used, while NIPC utilizes a set of orthogonal polynomials, such as Hermite polynomials.

5.3.3 Simulations

All simulations were performed using a multi-degree of freedom ship motion code. For the purposes of this research, an approximately linear variant of the simulation tool capable of modeling parametric roll was used. All simulations were setup and performed non-dimensionally. Simulations were based upon the Series 60 container ship model, with a block coefficient of 0.70 in regular seas with sinusoidal waveforms. Two separate simulations were set up and performed that serve as the raw data that will be approximated using these NIPC methods. Within these sets of simulations, all parameters are kept constant across runs, except for the wave frequency, ship speed, and heading angle (with respect to the waves). To generate specific cases for simulation, Gaussian distributions are used for each of the three input parameters.

For the first set of simulations the goal was to produce parametric roll in the majority of the individual simulations, but for this case higher roll amplitudes and an increased probability of capsize was expected. For these simulations the mean values of the input parameters were chosen to be 0.4 for the non-dimensional frequency, 0.065 for the Froude number, with the mean heading angle (180°) kept the same. These values represent a wave period of 11.17 s, and a speed of 5.53 knots and correspond to a wavelength to ship length ratio of 1 and a ratio of encounter frequency to the natural roll frequency of 1.75, very like the classic parametric roll cases seen in Paulling (2006). The standard deviations for Froude number, wave frequency, and heading angle were 0.01 (0.85 knots), 0.02 (.004476 Hz), and 5°, respectively.

For the second set of simulations, the mean values of these parameters were chosen so that parametric roll would generally occur, but capsize would not. This resulted in a mean non-dimensional frequency of 0.3, a mean Froude number of 0.2, and a mean heading angle of 180°, corresponding to a head sea. These values represent a wave period of 14.9 s and a speed of 17 knots and correspond to those used in previous studies (France et al, 2001; 2003). For the geometry in question, these parameters correspond to a ratio of wavelength to ship length of 1.8, and a ratio of the encounter frequency to the natural roll frequency of approximately 1.9. The standard deviations of the distributions for Froude number, wave frequency, and heading angle were 0.04 (3.4 knots), 0.04 (.008953 Hz), and 5° respectively. An example of the time histories produced numerically from the each set of parametric roll simulations can be seen in Figures 5.1 and 5.2 from the corresponding simulation sets, respectively.

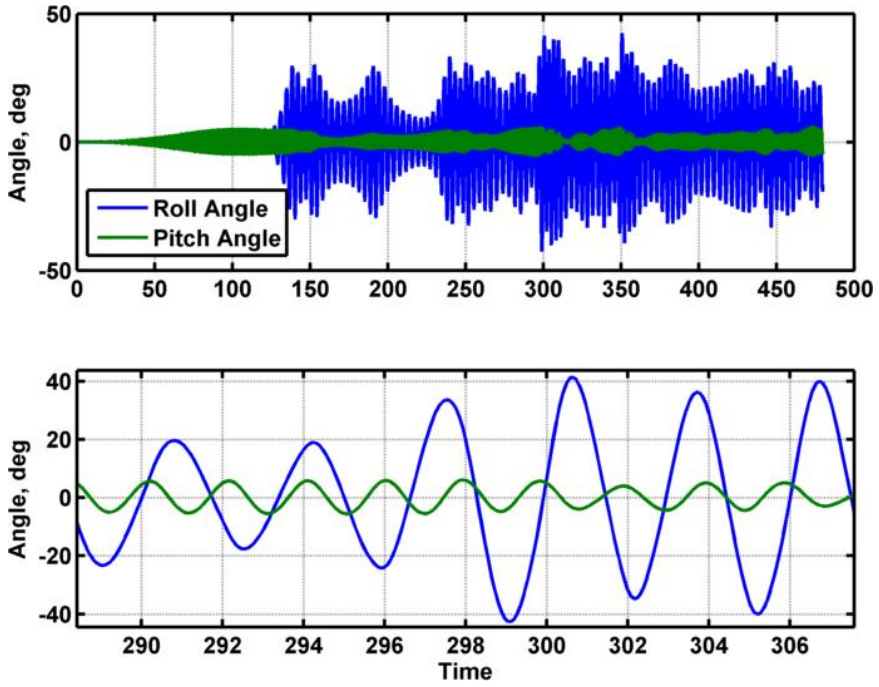


Figure 5.1: Example parametric roll trajectory from simulation from the first, higher amplitude, set of simulations.

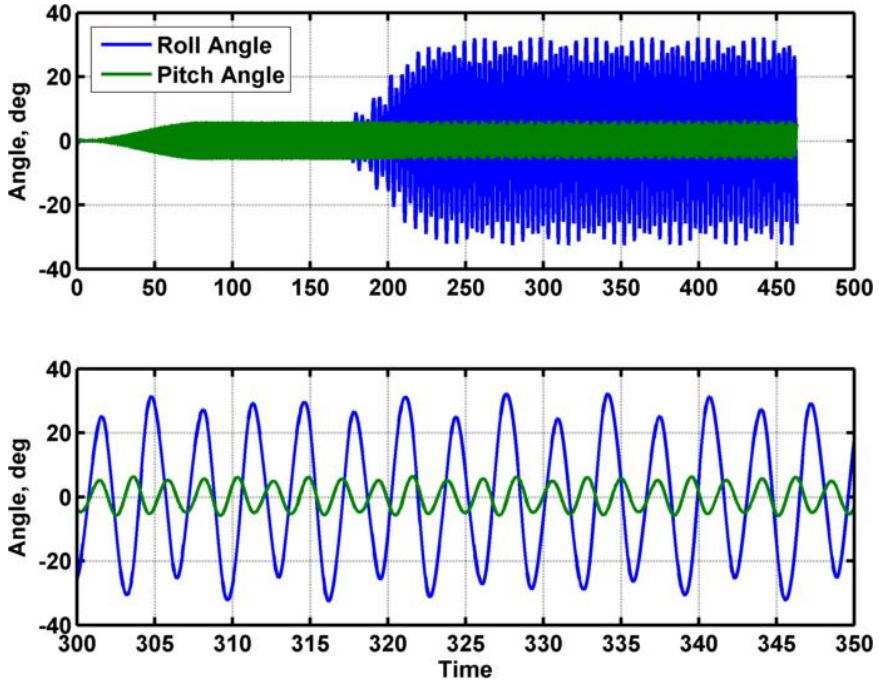


Figure 5.2: Example parametric roll trajectory from simulation from the second, lower amplitude, set of simulations.

5.4 Results & Discussion

5.4.1 NIPC Collocation Method

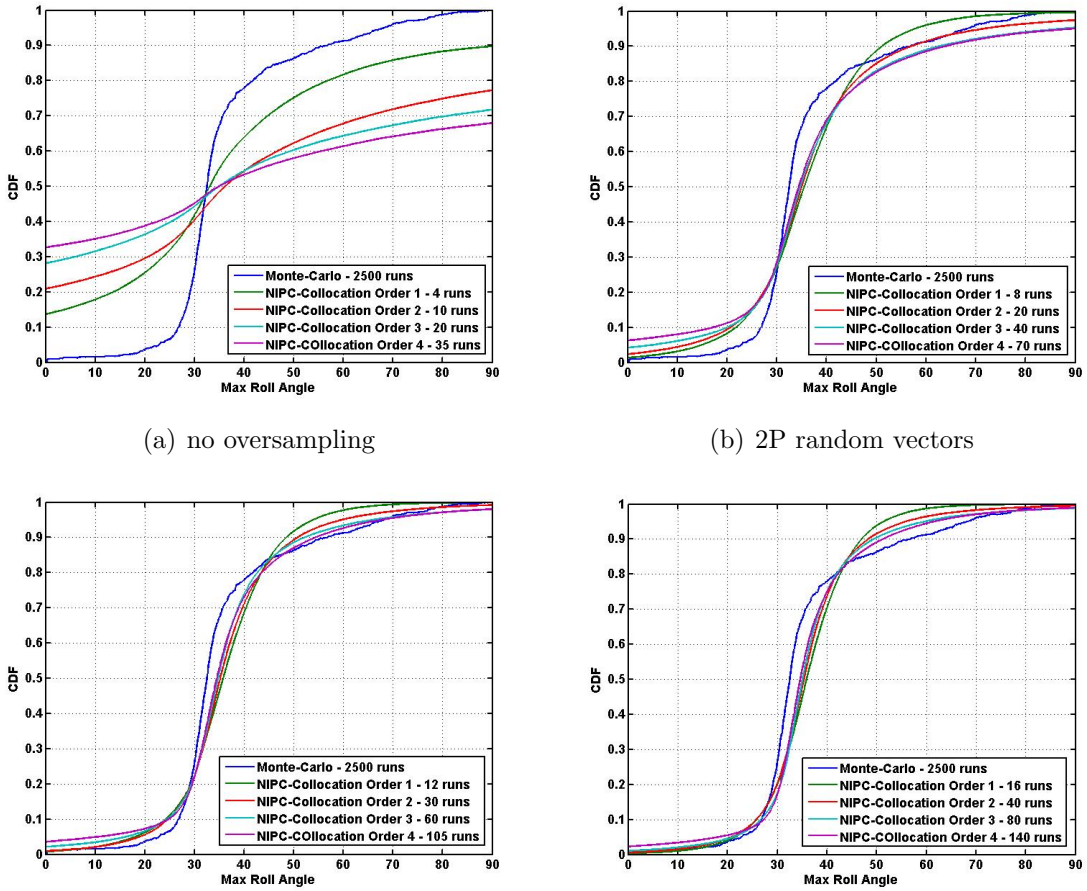
The first, generally higher amplitude parametric roll set of simulations, has twenty five hundred individual simulations which were generated using numerical simulation according to the previously discussed settings and model geometry. These results were sorted based upon the occurrence of capsize. The non-capsize runs had an average run time of approximately 4.92 hours while the capsize runs took approximately 4.12 hours per run. These runs were performed by 64 AMD Opteron series 6276 (2.3 GHz) processors with 2 GB of ram per core. There were 1,664 runs out of 2,500 total runs that did not capsize. To be concise, and due to the large number of capsize runs which capsized, only the non-capsize runs are considered for analysis with this data set. The relative effects of inclusion of capsize cases is considered later in this section for the lower amplitude parametric roll data set.

The collocation method was used to approximate the CDF of the maximum roll angles that occurred in the simulations. This was performed for different orders of NIPC polynomials (first order through fourth order). Also varied was the number of sample runs used to calculate the CDF. Additionally, the calculations to determine the NIPC coefficients were performed for 100 different sets of independent simulations so that 100 different approximations of the CDF are produced. The resulting mean CDFs is shown in Figure 5.3 as well as the mean resulting errors shown in Figure 5.4. Within Figure 5.3, the first subplot (5.3a) shows CDFs calculated using NIPC polynomial of orders one through four with the number of sampling points equal to the total number of polynomial terms, P, as determined by Equation 5.2. This gives P=4 runs for the first order calculation, P=10 runs for the second order calculation, P=20 runs for the third order calculation, and finally P=35 runs for the fourth order calculation. The

subplots 5.3b, 5.3c, and 5.3d used 2P, 3P, and 4P runs to calculate the CDF results, respectively.

Similar calculations were performed for the lower amplitude parametric roll set of simulations. However, for this data set two different scenarios were considered. The first scenario is analogous to the treatment of the non-parametric roll data set. Here, calculations were performed based only upon those simulations which entered parametric roll, but did not capsize (about 1% did not reach parametric roll; about 5% capsized). The CDFs calculated from this subset of the data are shown in Figure 5.5; errors are shown in Figure 5.8. In the second scenario, every simulation was retained, including those which capsized or did not enter into parametric role. Thus, the underlying random process is very nonlinear, with sudden transitions in behavior. The CDFs calculated from the complete set of low amplitude parametric roll data are shown in Figure 5.6; errors are shown in Figure 5.9. Note that these calculations were performed in the same manner as those presented for the large amplitude parametric roll data.

The same conclusions can be drawn from this data set as could be drawn from the higher amplitude parametric roll data set. With this data set, however, we perform an additional analysis into the effect of strong nonlinearities in the underlying random process, as such nonlinearities are present in the full data set, but are neglected in the subset of the data that only considers parametric roll without capsize. In the former case, the underlying function is not continuous rather it includes bifurcations where the maximum roll angles jump to a value of 90° once capsize is initiated. For the data set shown in Figure 5.6, which includes capsize and non-parametric roll cases, there is weaker agreement between the Monte Carlo result and the CDFs produced by the collocation method, in contrast to the excellent agreement seen in the parametric roll subset of data in Figure 5.5. That this affect is due to the nonlinearities associated with capsize is made apparent by Figures 5.7 and 5.10, which show comparable CDF



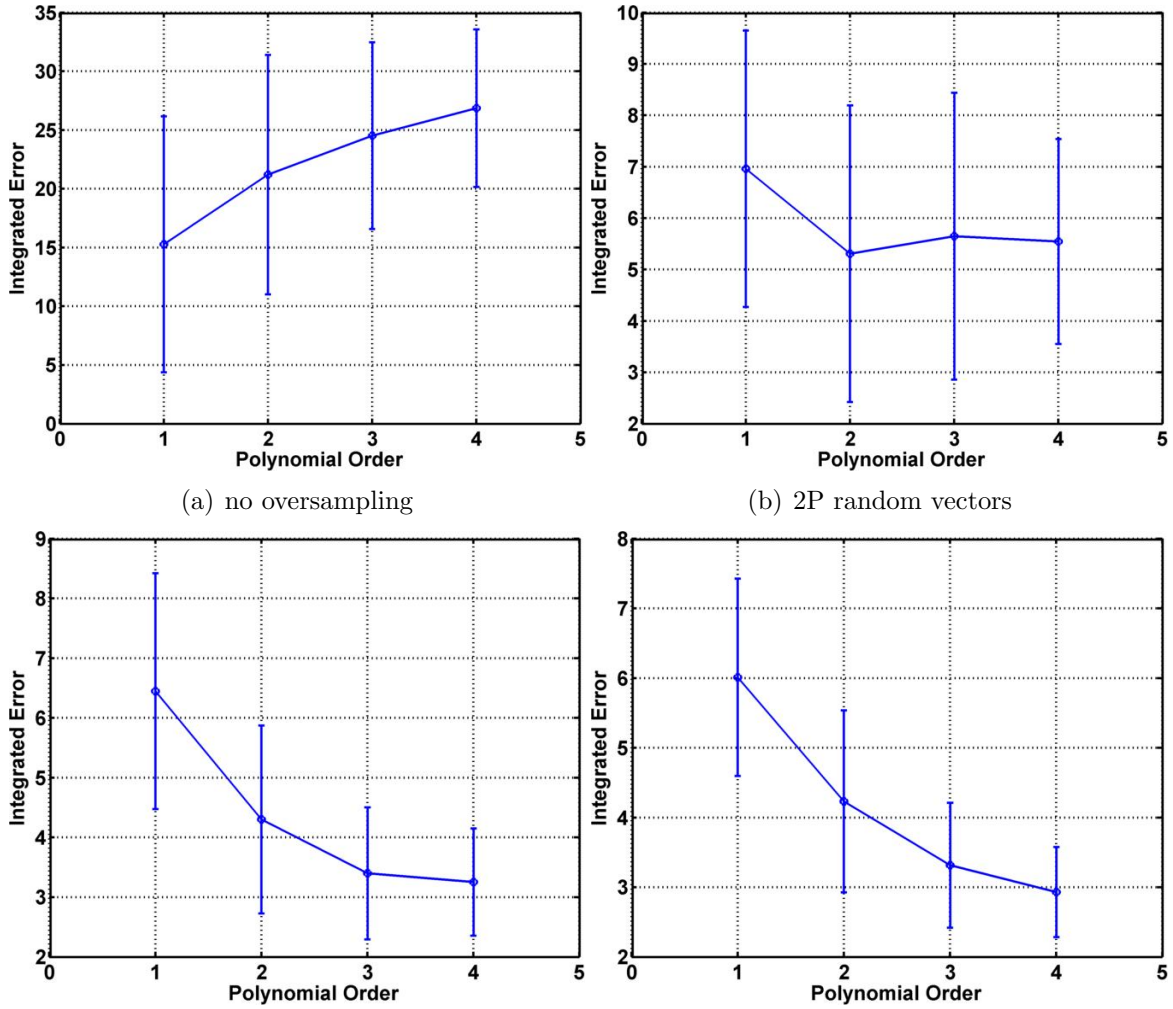
(a) no oversampling

(b) 2P random vectors

(c) 3P random vectors

(d) 4P random vectors

Figure 5.3: Comparison of expected collocation method conditional CDFs for the higher amplitude parametric roll data set, rejecting those runs which capsized. These results are based upon 100 randomly selected subsets of the data for polynomial orders of 1-4 with the Monte Carlo CDF for a) no oversampling (P random vectors), b) 2P random vectors, c) 3P random vectors, d) 4P random vectors. Increasing the number of random vectors using over P dramatically improves the accuracy of the method.



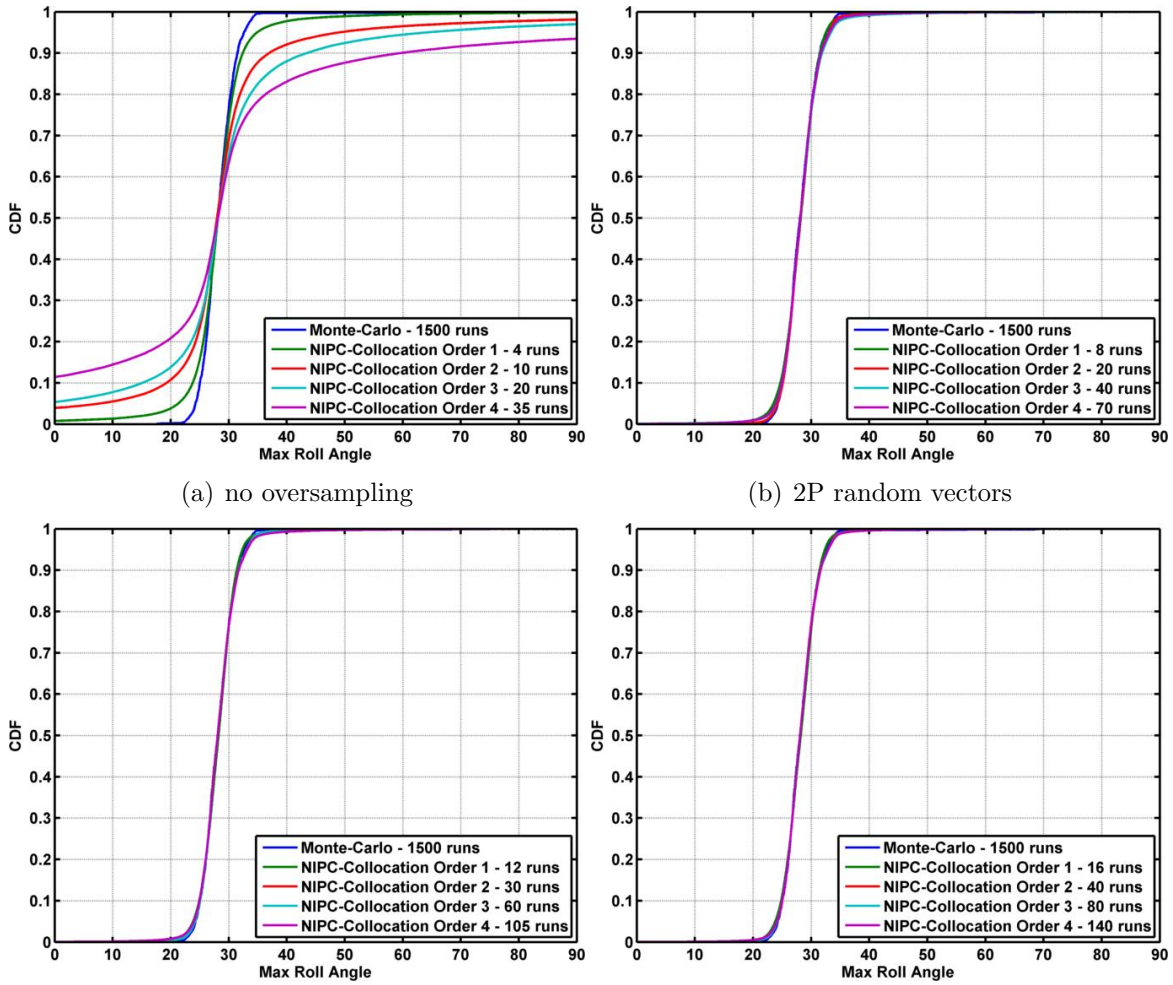
(a) no oversampling

(b) 2P random vectors

(c) 3P random vectors

(d) 4P random vectors

Figure 5.4: Expected (based upon 100 randomly selected subsets of the data) integrated error for the conditional collocation method CDFs for the higher amplitude parametric roll data set with respect to the conditional Monte Carlo CDF as a function of polynomial order for a) no oversampling (P random vectors), b) 2P random vectors, c) 3P random vectors, d) 4P random vectors. Increasing the number of random vectors using over P dramatically improves the accuracy of the method. Error bars represent one standard deviation.



(a) no oversampling

(b) 2P random vectors

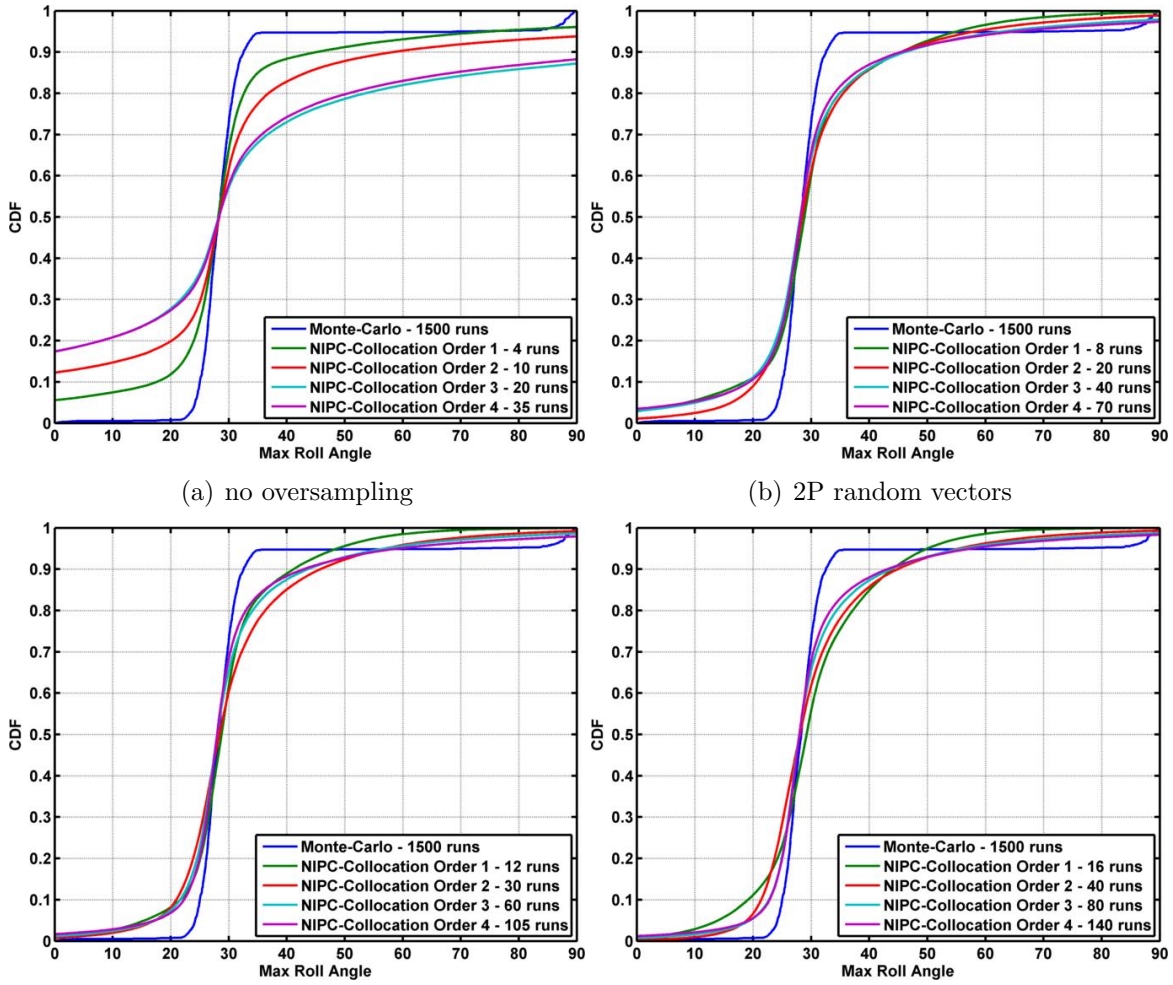
(c) 3P random vectors

(d) 4P random vectors

Figure 5.5: Comparison of expected collocation method conditional CDFs for a subset of the lower amplitude parametric roll data set (keeping only those runs where parametric roll occurs, but capsize does not) based upon 100 randomly selected subsets of the data for polynomial orders of 1-4 with the Monte Carlo CDF for a) no oversampling (P random vectors), b) 2P random vectors, c) 3P random vectors, d) 4P random vectors. Increasing the number of random vectors using over P dramatically improves the accuracy of the method.

and error results respectively when only the capsize cases are neglected. These results are very similar to those shown in Figures 5.5 and 5.8. This behavior is to be expected, as the collocation method relies on approximating the underlying random process as a polynomial. As a result, very high orders would be required to accurately approximate strongly nonlinear phenomena and it is difficult to capture bifurcations. In contrast, it should be noted that the errors shown in Figure 5.8 decrease significantly with increasing polynomial order.

In Figure 5.11, a comparison of the first and second (as represented by the standard deviation) statistical moments calculated for the second conditional capsize part of the data set are shown. Four different estimates of the moments are shown in these figures. The first of these (shown in blue) is the estimate derived directly from the NIPC approximation of the random process via Equation 5.3 and 5.4. The second (shown in green) is generated by evaluating NIPC polynomials for a large number of points drawn from the previously described input distributions. The third estimate (shown in red) is similar, except that the input distributions were modified to exclude values which would lead to capsize. In other words, this estimation represents conditional moments assuming that capsize did not occur. The final estimate (represented by a dashed black line) also represents the conditional moments (again assumping that capsize did not occur) but is based upon the results of the original Monte Carlo analysis of the ship motion code, prior to the NIPC. This is analogous to the population from which the input data for the NIPC analysis was drawn.
Focusing upon the mean values, it is clear that there is an offset between the mean value estimated by the NIPC calculation and that originating from the Monte Carlo analysis of the underlying random process. The origin of this is made clear by the two separate Monte Carlo type analyses of the NIPC polynomial (the second and third estimates described above). This offset is present when the full input parameter



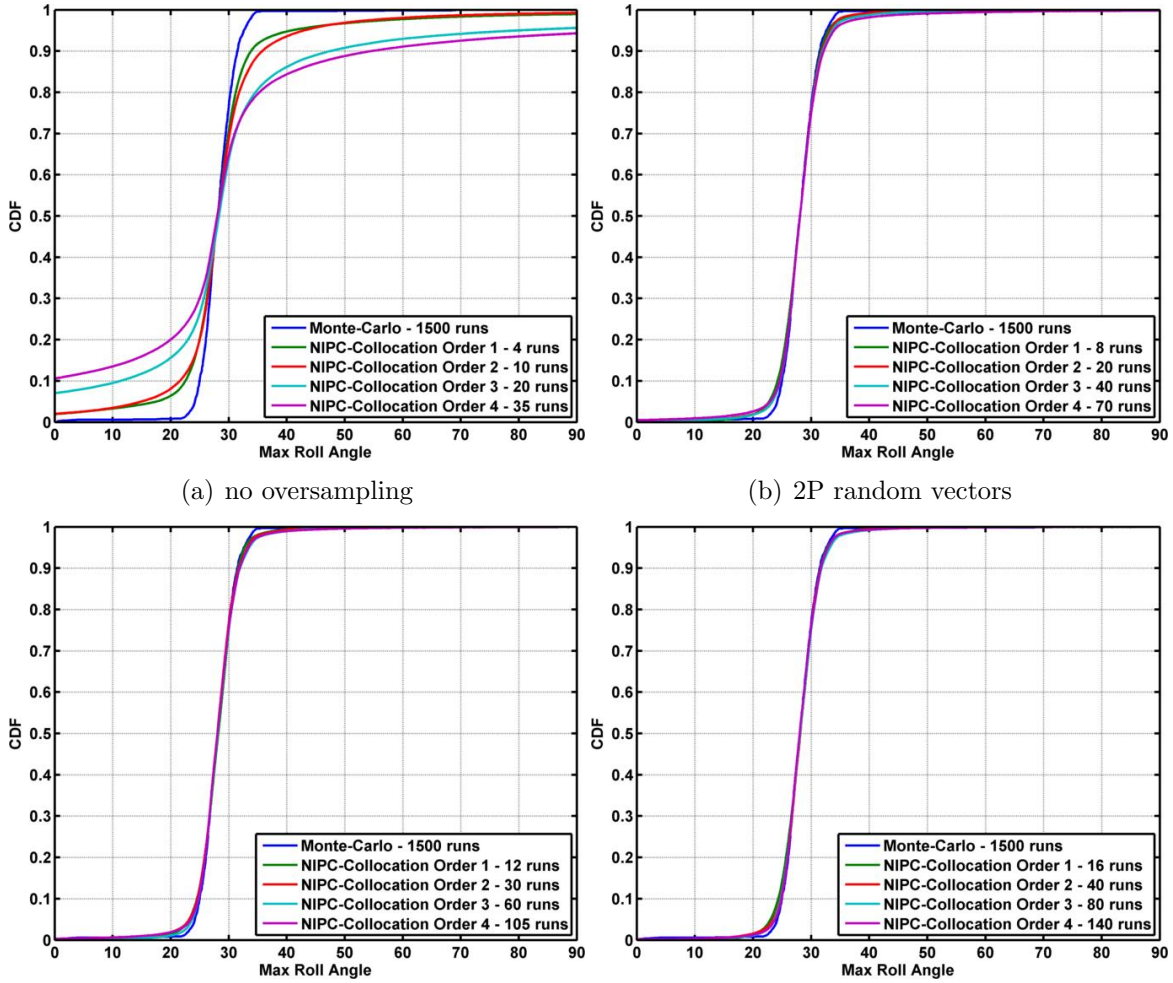
(a) no oversampling

(b) 2P random vectors

(c) 3P random vectors

(d) 4P random vectors

Figure 5.6: Comparison of expected collocation method CDFs (using the entirety of the lower amplitude parametric roll data set) based upon 100 randomly selected subsets of the data for polynomial orders of 1-4 with the Monte Carlo CDF for a) no oversampling (P random vectors), b) 2P random vectors, c) 3P random vectors, d) 4P random vectors. Increasing the number of random vectors using over P dramatically improves the accuracy of the method.



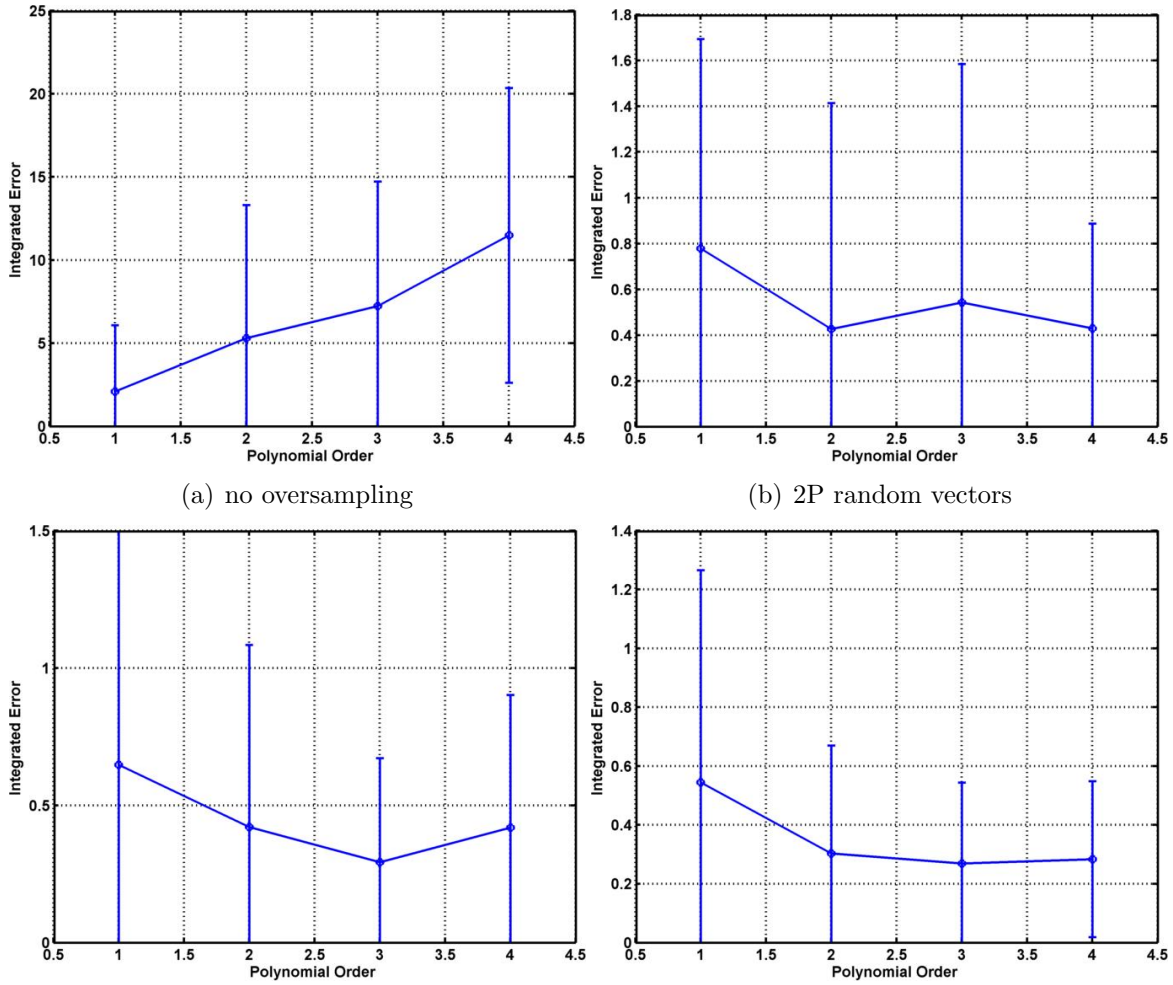
(a) no oversampling

(b) 2P random vectors

(c) 3P random vectors

(d) 4P random vectors

Figure 5.7: Comparison of expected collocation method CDFs (using the lower amplitude parametric roll data set without the capsize runs) based upon 100 randomly selected subsets of the data for polynomial orders of 1-4 with the Monte Carlo CDF for a) no oversampling (P random vectors), b) 2P random vectors, c) 3P random vectors, d) 4P random vectors. Increasing the number of random vectors using over P dramatically improves the accuracy of the method.



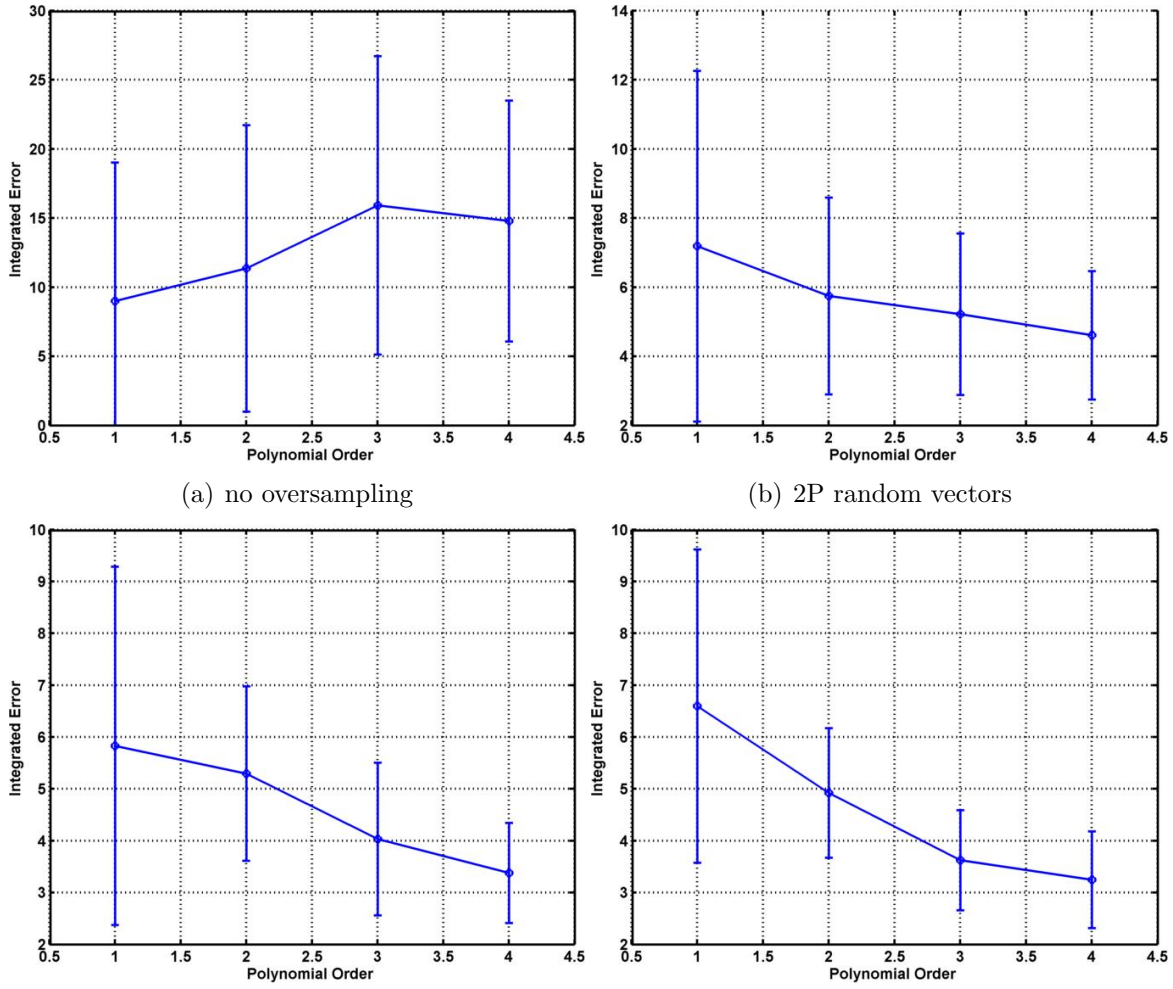
(a) no oversampling

(b) 2P random vectors

(c) 3P random vectors

(d) 4P random vectors

Figure 5.8: Expected (based upon 100 randomly selected subsets of the data) integrated error for the conditional collocation method CDFs calculated based upon a subset of the lower amplitude parametric roll data with respect to the conditional Monte Carlo CDF as a function of polynomial order for a) no oversampling (P random vectors), b) 2P random vectors, c) 3P random vectors, d) 4P random vectors. Increasing the number of random vectors using over P dramatically improves the accuracy of the method. Error bars represent one standard deviation.



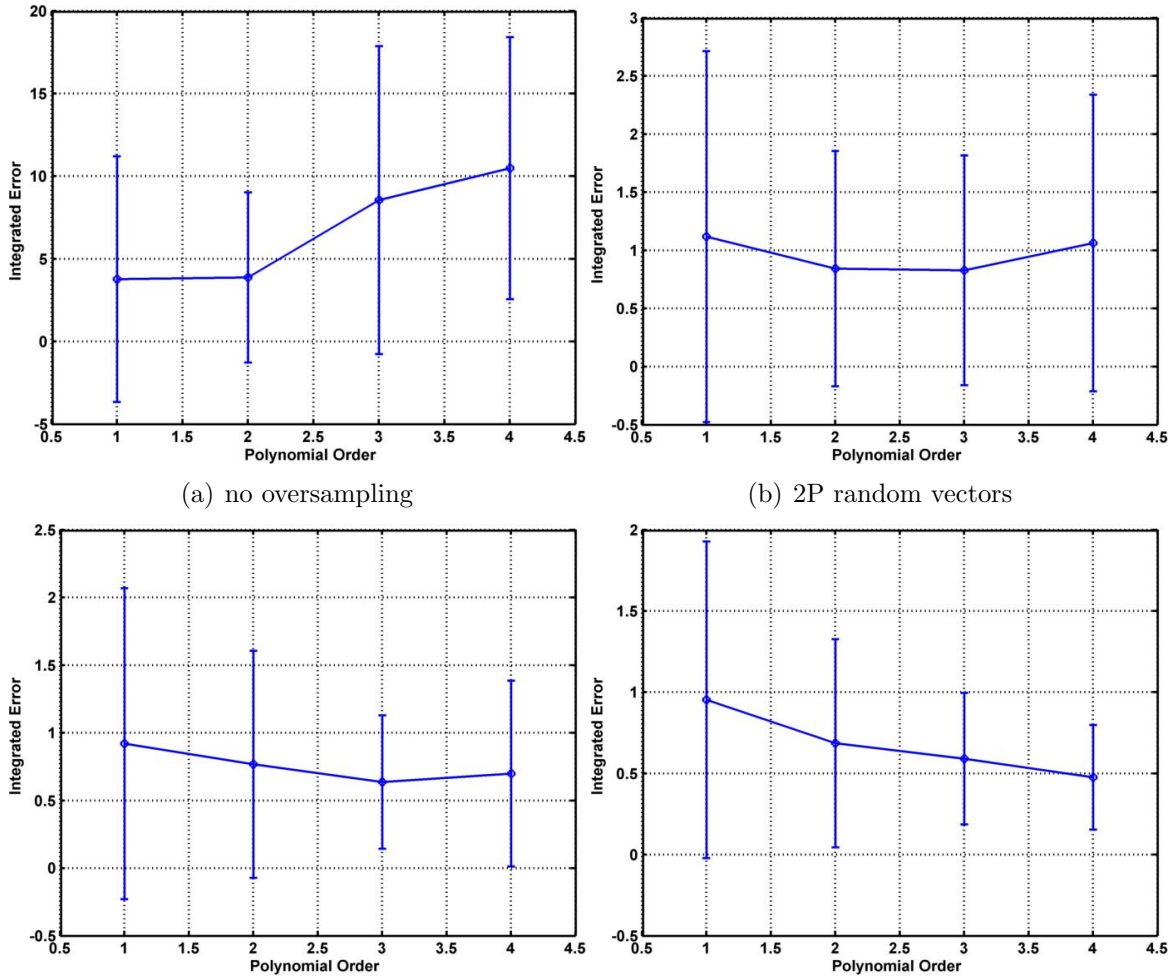
(a) no oversampling

(b) 2P random vectors

(c) 3P random vectors

(d) 4P random vectors

Figure 5.9: Expected (based upon 100 randomly selected subsets of the data) integrated error for the collocation method CDFs based upon the complete set of lower amplitude parametric roll data with respect to the Monte Carlo CDF as a function of polynomial order for a) no oversampling (P random vectors), b) 2P random vectors, c) 3P random vectors, d) 4P random vectors. Increasing the number of random vectors using over P dramatically improves the accuracy of the method. Error bars represent one standard deviation.



(a) no oversampling

(b) 2P random vectors

(c) 3P random vectors

(d) 4P random vectors

Figure 5.10: Expected (based upon 100 randomly selected subsets of the data) integrated error for the collocation method CDFs based upon the complete set of non-capsize parametric roll data with respect to the Monte Carlo CDF as a function of polynomial order for a) no oversampling (P random vectors), b) 2P random vectors, c) 3P random vectors, d) 4P random vectors. Increasing the number of random vectors using over P dramatically improves the accuracy of the method. Error bars represent one standard deviation.

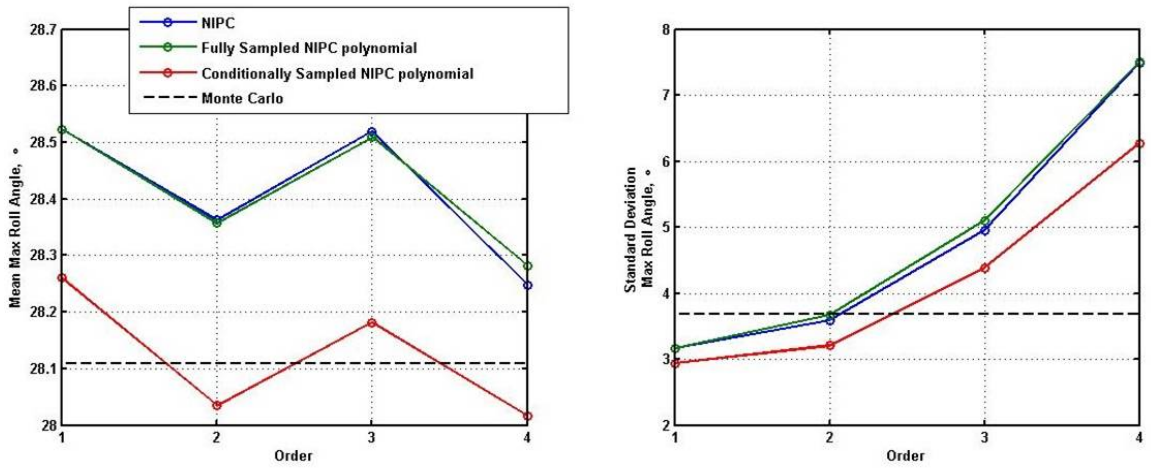


Figure 5.11: Statistical moments calculated using both NIPC and the Monte Carlo method.

space is sampled, it is absent when portion of the input parameter space which leads to capsize is included. The offset is present due to the fact that Equation 5.3 and 5.4 inherently sample the entirety of the input parameter space the equations provide no allowance for conditional probabilities, for the sampling of a subspace. While the collocation approach to NIPC can still be used to evaluate this type of problem (as evidenced by the conditional CDFs shown in Figure 5.7), Equation 5.3 and 5.4 cannot necessarily be used to evaluate statistical moments from the polynomial fitted by the NIPC method. Alternate means of evaluating moments from the fitted polynomial may instead be necessary. If the boundary (in the input parameter space) between allowed and disallowed states is well understood, this is easily accomplished, in other circumstances this may be more difficult.

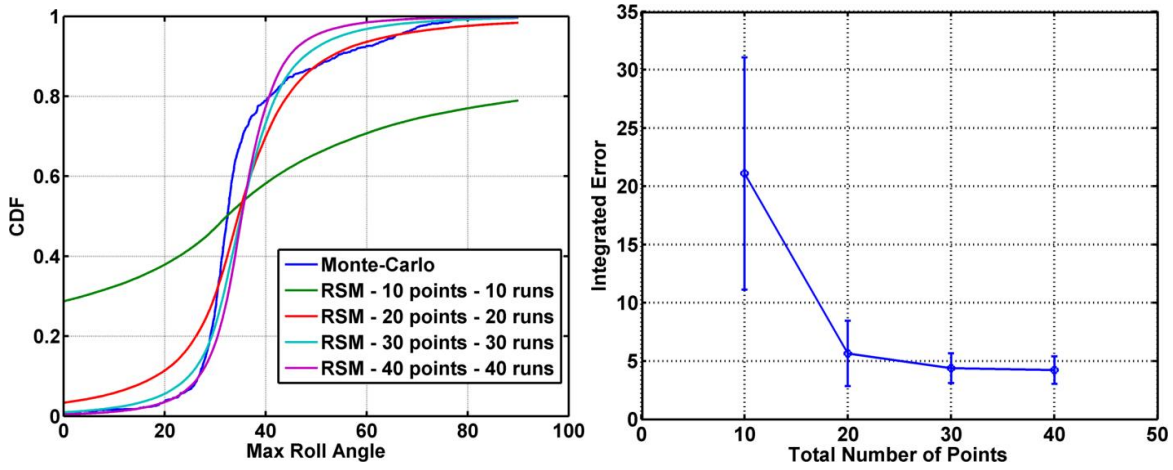
5.4.2 Response Surface Method

For each set of data, the CDF of the maximum roll angle was also estimated using the second order response surface method, in order to provide a basis for comparison for the NIPC collocation results. As was the case for the collocation data, the sample

	2nd Order NIPC Collocation	2nd Order RSM
10 points	11.35	9.77
20 points	5.75	5.74
30 points	5.29	5.04
40 points	4.91	4.89

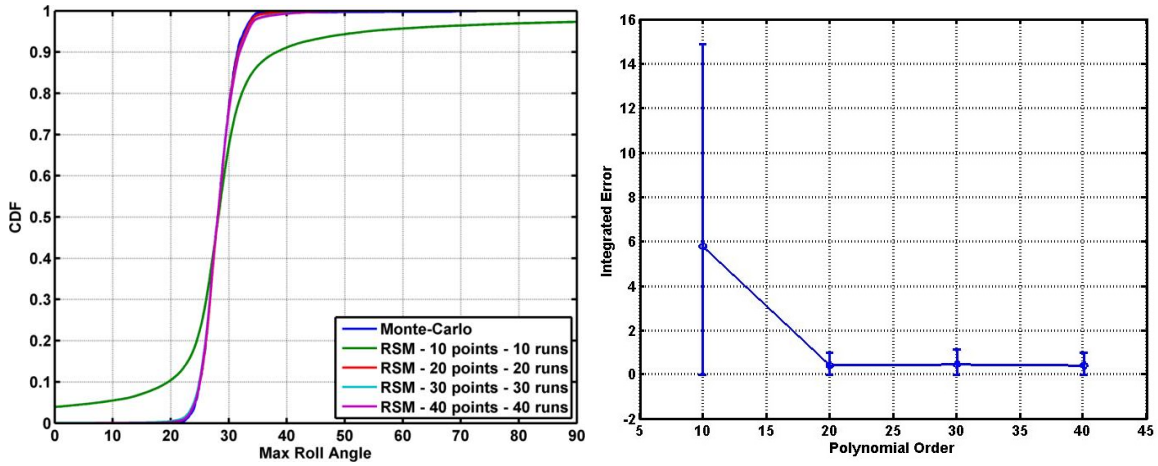
Table 5.1: Comparison of integrated errors for the 2nd order NIPC collocation method, and the 2nd order response surface method for 10, 20, 30, and 40 randomly selected sample points in the full parametric roll data set. In general, the two methods exhibit comparable performance. All differences are statistically insignificant.

points were randomly distributed, with CDFs being estimated based upon 10, 20, 30, and 40 sample points. Thus, the results for the response surface method are directly comparable to the second order collocation results, with P, 2P, 3P, and 4P oversampling. The primary difference between the two sets of results (NIPC and RSM) is that Hermite polynomials are used to represent the process with NIPC collocation, while single term polynomials are used with the response surface method. The expected CDF and the corresponding errors produced by the response surface method are shown in Figure 5.12 for the high amplitude parametric roll data, Figure 5.13 for the conditional parametric roll results and Figure 5.14 for the complete parametric roll results. From these results, it is apparent that the results from the two methods are quite similar, with both methods displaying approximately the same computational efficiency. This is not surprising, since fundamentally the difference between the two methods is the choice of fitted polynomial used, as discussed previously. To clarify this, the integrated errors of the corresponding response surface and collocation methods for the full set of parametric roll data are compared in Table 5.1.



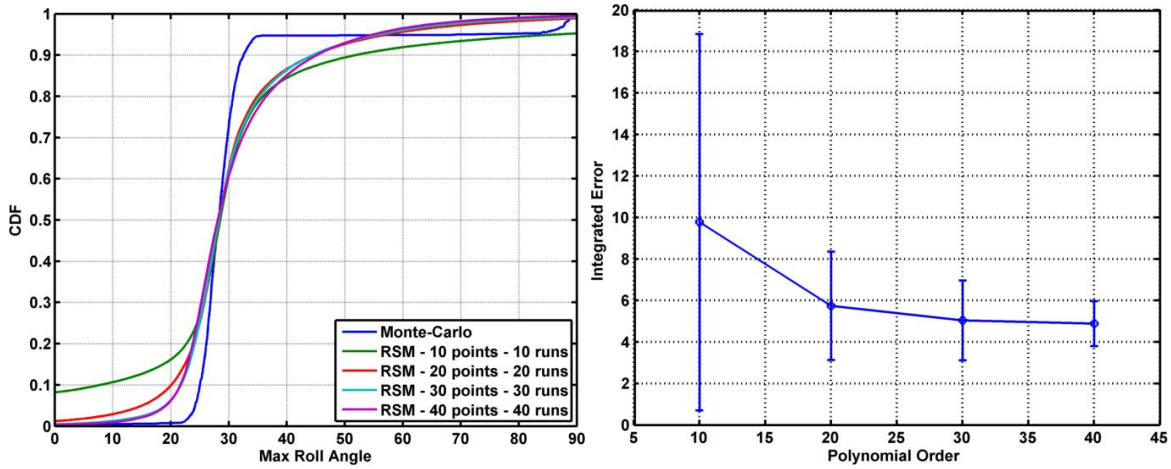
(a) (b)

Figure 5.12: a) Comparison of the CDFs generated using the response surface method for the high amplitude parametric roll data to the Monte Carlo CDF for various numbers of data points from the conditional data set (no capsize) b) Expected integrated error with respect to the Monte Carlo CDF as a function of the number of points used.



(a) (b)

Figure 5.13: a) Comparison of the CDFs generated using the response surface method to the Monte Carlo CDF for various numbers of data points from the conditional low amplitude parametric roll data (all runs in parametric roll; none capsize) b) Expected integrated error with respect to the Monte Carlo CDF as a function of the number of points used.



(a) (b)

Figure 5.14: a) Comparison of the CDFs generated using the response surface method to the Monte Carlo CDF for various numbers of data points from the complete low amplitude parametric roll data set b) Expected integrated error with respect to the Monte Carlo CDF as a function of the number of points used.

5.5 Conclusions

In the present study, NIPC collocation and response surface methods were used to estimate the CDF of maximum roll angles in high and low amplitude parametric roll simulations performed using numerical simulation. The resulting estimates of the CDF of the maximum roll angle of the simulations using the NIPC collocation method are compared to those calculated via a Monte Carlo simulation and predicted via the response surface method. In the absence of strong nonlinearities in the underlying process, the NIPC collocation method was able to approximate the Monte Carlo simulation result with a high degree of accuracy. However, it was discovered that in general, this required at least 2P oversampling to achieve accurate results which confirms Hosder et al. (2006) and Wu et al. (2011) findings. Additionally, it was found that variability exists with the NIPC collocation method and response surface method, depending upon the (random) choice of simulation points.

In the case where bifurcations such as capsize are present, substantially increased errors are encountered. However, NIPC demonstrated the ability to function for these processes as well, provided a sufficiently high order polynomial is used. In some cases however, if this order is too high, it may require less data points to simply perform a Monte Carlo analysis. Outside of this scenario, however, NIPC was able to demonstrate substantial computational savings compared to the Monte Carlo method. For example, evaluation of the highest order NIPC considered (4th order) the NIPC calculations took 50 core-hours at the suggested 2P level of oversampling for the low amplitude parametric roll data set, see Figure 6b. In contrast, the corresponding MC analysis required approximately 1800 core-hours to complete the MC results shown on the same Figure 6b. Comparing the results for the response surface method to those for the NIPC collocation method, it is shown that the response surface results were very similar. As discussed previously, this is unsurprising, since both represent least squares fits of polynomials to randomly scattered data.

The demonstrated ability of NIPC to derive meaningful statistical distributions from a small number of samples is very useful to a large number of applications and fields. One of the most noteworthy applications, as disussed previously, is uncertainty analysis. This makes NIPC very useful for validation, as uncertainty quantification is generally required to properly compare experimental and model values for system response quantification (Oberkampf & Roy, 2010). This is true for the chaotic systems whose validation is discussed in the previous chapters, but also for any simulation or experiment where input uncertainties cannot easily be propagated to the final result.

In future work it will be of interest to see if there is still sufficient benefit for problems where where bifurcations cannot be avoided, e.g. when capsize is specifically of in-

terest. Also, further study is warranted toward better understanding the uncertainty associated with the use of collocation-based NIPC compared with the uncertainty associated with the Monte Carlo method. This is particularly true when limited data is available due to numerical or experimental constraints. A method for identifying when a sufficiently high order polynomial has been fit to the random process when utilizing NIPC would also be quite valuable. Ideally, a rule governing polynomial order would be determined, so that a priori knowledge by the user is not required. Even without such an improvement, however, when used with appropriate care the NIPC algorithm has demonstrated a noteworthy ability to derive statistical information from highly nonlinear models at substantially reduced computational cost. This makes the use of computationally expensive models much more feasible for stochastic applications.

CHAPTER VI

Design of a Controller for Autonomous Vessel Recovery Utilizing the Prediction of Host Vessel Motions

6.1 Introduction

Model based control represents an important application of nonlinear dynamics modeling and is relevant to large variety of fields, including ship dynamics. For many applications, the development of a controller represents the final step in the process of applying a model in the real world. Much like validation, the process of controller derivation can be significantly complicated when the dynamics of the model (and the underlying physical system) are nonlinear. Here, a ship dynamics model, combined with nonlinear ship motion prediction algorithms, are utilized to develop a controller for the automation of the recovery of Unmanned Surface Vehicles (USVs). USVs are of increasing importance for both scientific and military applications. The US Navy sees such craft contributing to a vast array of missions, ranging from mine countermeasures (MCM) to anti-submarine warfare (ASW) to electronic warfare (EW)

This chapter is derived in large part from Cooper and McCue, 2015.

(US Navy, 2007). These vessels have significant advantages over manned craft in endurance, cost, and expendability (US Navy, 2007). Although some such vessels are remotely operated, there are also vessels in use and/or under development which are fully autonomous between launch and recovery, or at minimum, need only high level input from an operator (e.g., waypoints) (Manley, 2008; Roberts and Sutton, 2006). Much work has already been performed regarding the development of algorithms for the controlled navigation of such vessels, ranging from relatively simple approaches such as proportional-integral-derivative controllers (Caccia et al, 2008) to much more sophisticated adaptive controllers (Leonessa et al, 2006; Breivik et al, 2008). Many such algorithms operate under the paradigm of defining a path for the USV to follow, with the controller acting to follow the path as closely as practical (Bibuli et al, 2009).

Many, if not most, of these USVs are designed to operate from a larger manned craft and must be recovered at the end of a mission or for maintenance and re-fueling. Recovery events are likely to be frequent by necessity. The benefits associated with launch and recovery of aerial and marine vehicles have been recognized for some time. Eugene Ely first flew off the deck of a ship not long after the Wright brothers first flight, and large vessels have been host to smaller vessels since the dawn of maritime travel. Scientific study toward development of robust, real-time pilot cueing aids for detecting vessel quiescence was established by researchers such as O'Reilly (1987), Ferrier et al (1998, 2000a, 2000b), and Colwell (2002) to name but a few. Yet, to date, no truly comprehensive solution for the launch and recovery of USVs under a wide range of sea states has been developed (Roberts and Sutton, 2006), rather, there are many vehicle specific systems in use. While davits, cranes, and stern recovery systems have been used, many of these require manning in the loop. This research is focused upon pairing short term motion prediction with a robust controller toward automated recovery approaches to reduce crew load and exposure in a manner which

may be generalized to a variety of different recovery systems.

The purpose of this work is to demonstrate a control algorithm for automated recovery of USVs which utilizes ship motion prediction (applied to the host ship) to aid in the control of the USV during its recovery maneuvering. This controller assumes a stern ramp recovery, and acts to maneuver the USV into the recovery ship. This is done by defining a priori a trajectory the USV is to follow during the recovery maneuver, with the controller acting to execute the maneuver. In severe sea states, this entry maneuver may only be safe and effective during relatively short windows of time, such as when the recovery ship is at small roll and pitch angles. As a result, it is necessary to tie a ship motion prediction capability into the controller, so that the entry maneuver can be timed to correspond to optimal conditions. Ship motion prediction is an active area of research. The complexity of ship roll and pitch motions makes prediction difficult, and there is not usually one solution that works the best under all circumstances. As a result, two separate approaches are considered herein, one based upon artificial neural networks and another based upon the identification of nearest neighbors in past trajectory data.

Neural networks have been successful in predicting and/or controlling ship motion (see, for example, Haddara and Xu (1999), Liut et al (2000), Xing and McCue (2009), (2010) and Hess et al., (2006a), (2006b)) and have been previously investigated to aid in the recovery of unmanned aerial vehicles on naval type ships (Khan et al, 2005). As a result, a neural network is used to predict the motion of the recovery ship, so that the entry maneuver by the USV can be timed so that entry corresponds into the recovery ship has an approximately zero pitch angle and a small roll angle. (Note this could similarly be defined for zero roll angle or zero pitch angle alone.) A separate predictor algorithm, based upon identification nearby ship states in previously recorded

ship behavior is also investigated following the approach discussed in McCue (2008). The ability of the predictors to meet the requirements of the derived controller were numerically tested using experimental David Taylor Model Basin (DTMB) model data for a notional destroyer model (Hayden et al, 2006) to simulate the behavior of a host/recovery vessel. Numerical simulations of the recovery procedure were performed based upon the behavior of the RibCraft USV, used as a sample autonomous vehicle due to the readily available vessel modeling performed by Sonnenburg (2012).

6.2 Theoretical Modeling

Before the details of the derived controllers can be discussed, the details of the theoretical model of the USV behavior used in this work must be discussed. Sonnenburg states that under many conditions the coupling between speed and course for USV type vessels is negligible (Sonnenburg, 2012). As a result, separate models for the speed and course behaviors of the USV were defined. For speed, a quadratic model is applied, so that the forward motion of the vessel is governed by:

$$\begin{aligned} \dot{u}(t) &= u(t)\left(au(t)+b\right)+\cos(\delta(t))T(t) \\ \dot{x}(t) &= u(t) \end{aligned} \tag{6.1}$$

where a is the quadratic drag coefficient, b is a linear drag coefficient, δ represents the angle at which thrust is applied, and T is the thrust applied to the USV. This assumes that steering is performed by thrust vectoring. If the angle at which thrust is applied is limited to small angles, then this equation can be approximated by:

$$\begin{aligned} \dot{u}(t) &= u(t)\left(au(t)+b\right)+T(t) \\ \dot{x}(t) &= u(t) \end{aligned} \tag{6.2}$$

This is also the equation governing the surge dynamics of the USV in the absence of thrust vectoring. The course behavior of the USV is modeled using the Nomoto model without side slip (Nomoto, et al, 1957, Sonnenburg, 2012). This model is given by:

$$\begin{aligned} \dot{r}(t) &= -\frac{r(t)}{\tau} + K\sin(\delta(t))T(t) \\ \dot{\theta}(t) &= r(t) \end{aligned} \tag{6.3}$$

where $r(t)$ represents turning rate, τ represents a time constant associated with the viscous loss of angular momentum (so that $1/\tau$ may be thought of as a rotational drag term), and θ represents the course angle of the USV (with respect to the course of the recovery ship).

However, in the present case, it is presumed important that the USV be located directly behind the recovery ship such that the USV has the same course as the recovery ship. As a result, it is necessary to add an additional equation to the model governing the y distance between the recovery ship and the USV. Including this equation, the complete system becomes:

$$\begin{aligned} \dot{r}(t) &= -\frac{r(t)}{\tau} + K\sin(\delta(t))T(t) \\ \dot{\theta}(t) &= r(t) \\ \dot{y}(t) &= -\sin(\theta(t))u(t) \end{aligned} \tag{6.4}$$

For the purpose of deriving a controller, however, it is assumed that the differences

in course angle are small, and therefore the system can be approximated as:

$$\begin{aligned}\dot{r}(t) &= -\frac{r(t)}{\tau} + K\delta(t)T(t) \\ \dot{\theta}(t) &= r(t) \\ \dot{y}(t) &= -u(t)\theta(t)\end{aligned} \tag{6.5}$$

Note that this last equation adds 1-way coupling between the speed and course behavior of the USV. Forward speed is not affected by the course, but because horizontal distance y has been added to the course subsystem, the course subsystem is affected by the forward speed of the vessel. Furthermore, in the absence of thrust vectoring, the influence of thrust on the steering dynamics is eliminated, so that:

$$\begin{aligned}\dot{r}(t) &= -\frac{r(t)}{\tau} + K\delta(t) \\ \dot{\theta}(t) &= r(t) \\ \dot{y}(t) &= -u(t)\theta(t)\end{aligned} \tag{6.6}$$

However, a similar form can be created for the case of thrust vectoring by defining:

$$\delta(t) = \frac{\delta'(t)}{T(t)} \tag{6.7}$$

The substitution of this expression for the thruster angle results in:

$$\begin{aligned}\dot{r}(t) &= -\frac{r(t)}{\tau} + K\delta'(t) \\ \dot{\theta}(t) &= r(t) \\ \dot{y}(t) &= -u(t)\theta(t)\end{aligned} \tag{6.8}$$

Thus, separate control laws can be implemented for both submodels (speed and course) with the caveat that modifications must be made to the course controller when thrust vectoring is utilized. The control laws discussed below are based upon

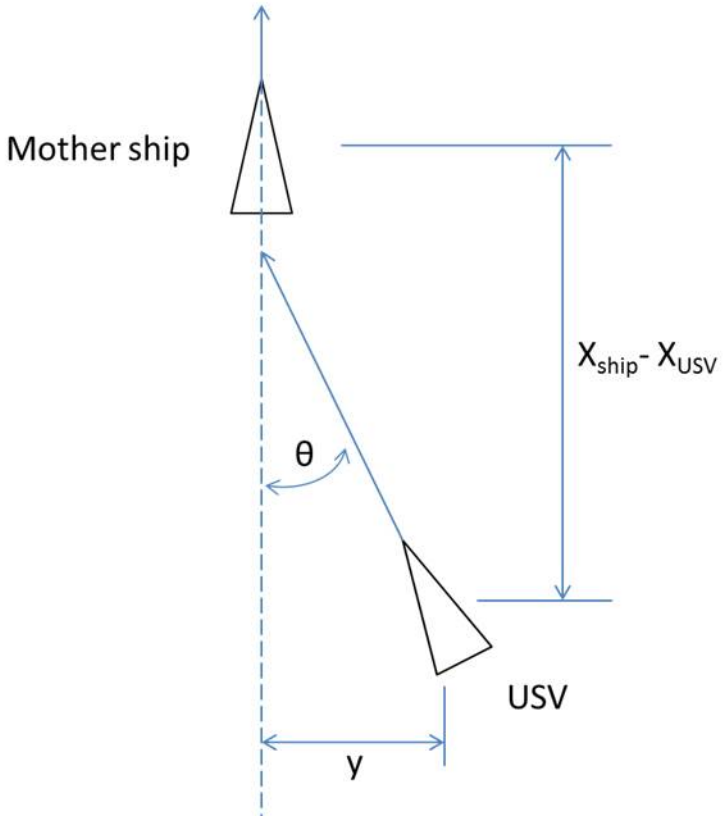


Figure 6.1: Coordinate system for the ship recovery controller.

the system described by Equation 6.8.

6.3 Controller Design

In this work, it is assumed that the USV recovery is to occur at the stern. The recovery ship is assumed to be moving forward at a nominally steady speed and course, with the USV approaching from the rear, as shown in Figure 6.1. The wave influence on the surge dynamics of the USV is not accounted for. The authors recognize this represents a nontrivial simplification that is perhaps, at least partially justified, by presuming the stern recovery will enable the USV seeing a reduced sea state as it travels in the wake of the host vessel. However, numerical results (shown later) indicate the controller to be relatively robust against such unmodeled dynamics. It should also

be noted that with appropriate sea forecasting and wake modeling, these dynamics can and should be implemented into a launch and recovery controller by changing the values of the physical parameters (e.g. drag coefficients) in the model and/or growing the model to represent the physical realities encountered in the wake. With this in mind, the trajectory followed by the recovery ship is defined mathematically as:

$$\begin{aligned} x_{ship}(t) &= Vt \\ y_{ship}(t) &= 0 \end{aligned} \tag{6.9}$$

where V represents the (assumed constant) speed of the recovery ship. While letting the position of the ship be zero at time equals zero allows for convenient derivation, it is not strictly necessary, and this arbitrary choice of zero location is chosen simply for convenience, representing a sort of initial point of the recovery maneuver. Since under more severe sea states the timing of this entry is important, the recovery maneuver is separated into two phases, each of which governed by its own controller. In the first phase, the waiting phase, the USV moves to a (moving) point at a specific distance directly behind the recovery vessel. The vessel then attempts to remain at this point relative to the recovery ship (in other words, it attempts to trail the ship at some specific distance). This point serves as the initial point for the actual entry maneuver, which is the second phase of the recovery maneuver, the entry phase. The advantage of defining the recovery maneuver into two phases is that straightforward and analytic desired trajectories can be predefined for each phase. For the first (waiting) phase, this desired USV trajectory is simply:

$$\begin{aligned} x_d(t) &= Vt - d \\ y_d(t) &= 0 \end{aligned} \tag{6.10}$$

Here, d represents the intended distance between the waiting USV (at the initial point) and the recovery ship and is henceforth referred to as the trailing distance. Defining the desired trajectory for the entry phase of the maneuver is more complex. In this case, the trajectory was designed to allow a smooth transition over a specified period of time between the trajectory followed by the USV during the waiting phase, and that of the recovery vessel. The requirements for such a trajectory are that the USV has the same velocity as the recovery ship at the start and end of the maneuver while closing the gap between the USV and the recovery ship over the course of the maneuver. Additionally, the desired velocity corresponding to the trajectory should be continuous. Based upon these requirements, a desired trajectory:

$$\begin{aligned} x_d(t) &= \left(\frac{d}{T} + V\right) t - \frac{d}{2\pi} \sin\left(\frac{2\pi t}{T}\right) \\ y_d(t) &= 0 \end{aligned} \tag{6.11}$$

was derived by inspection, with T representing the total time allotted for the execution of the entry phase of the maneuver. These trajectories are contrasted visually in Figure 6.2. Note, an alternative trajectory could be defined should it be desirable for the boat to overtake the host ship, for example, to ascend a stern ramp. Indeed, that would reduce the need for a relatively swift deceleration within the controller.

As discussed previously, the course and forward speed of a vessel can be decoupled in this model. This allows for separate controllers to be derived for the speed and course portions of the theoretical model, as shown in Figure 6.3. Unlike the speed controller, the objective of the course controller throughout the entirety of the recovery maneuver is to ensure that the USV remains directly behind the recovery vessel, and on the same course. Since the course subsystem is assumed to be linear (as discussed above) this can be accomplished via a simple linear quadratic regulator controller. For clarity,

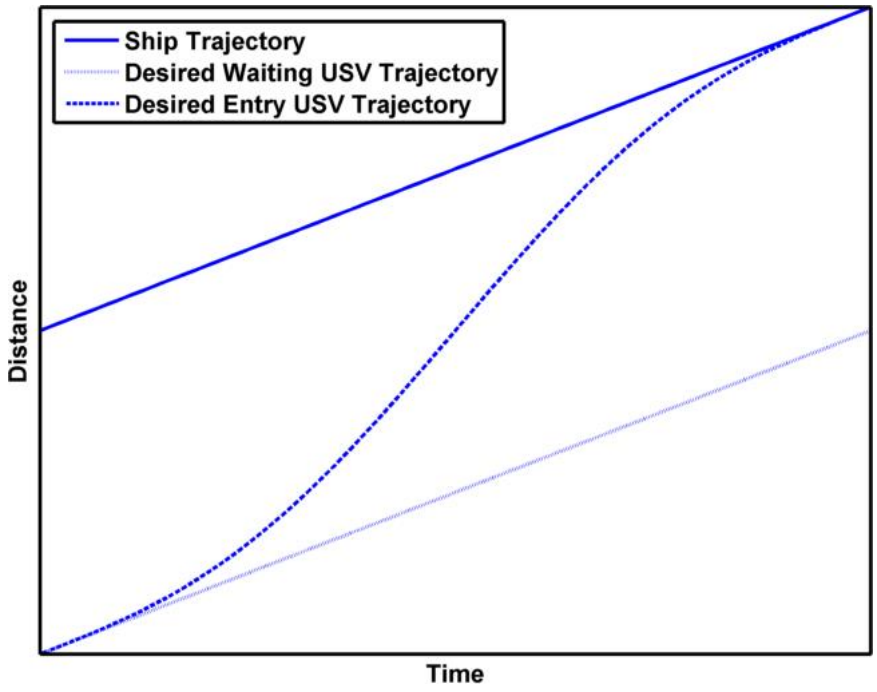


Figure 6.2: Illustration of hypothetical desired USV trajectories for both the waiting and entry phases of the controller. Note how in the entry phase the USV decelerates to the velocity of the recovery ship.

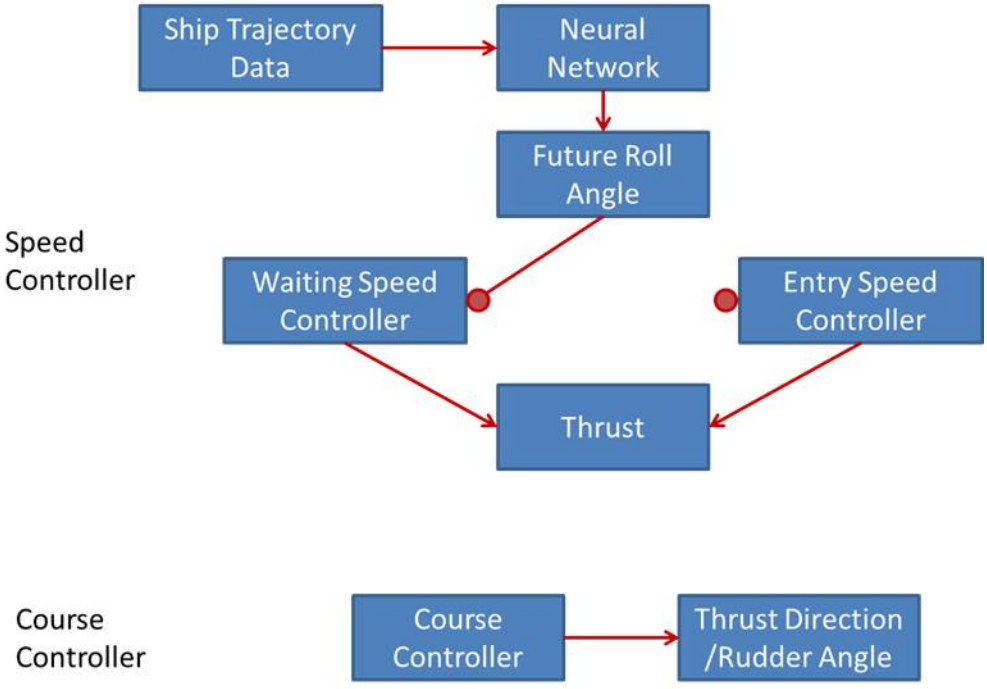


Figure 6.3: Schematic demonstrating the flow of information in the controllers. Not shown is the dependence of the course system on USV velocity.

the course subsystem is rewritten here as a matrix system of equations, so that:

$$\begin{bmatrix} \dot{y}(t) \\ \dot{\theta}(t) \\ \dot{r}(t) \end{bmatrix} = \begin{bmatrix} 0 & -u(t) & 0 \\ 0 & 0 & 1 \\ 0 & 0 & -\frac{1}{\tau} \end{bmatrix} \begin{bmatrix} y(t) \\ \theta(t) \\ r(t) \end{bmatrix} + KT(t) \begin{bmatrix} 0 \\ 0 \\ \delta(t) \end{bmatrix} \tag{6.12}$$

As a result, the optimal control law is given by:

$$\delta(t) = -\mathbf{R}^{-1}[0 \quad 0 \quad KT(t)]\mathbf{P}(t) \tag{6.13}$$

where $\mathbf{P}(t)$ is the solution to the algebraic Riccati equation, and $\mathbf{R}$ is the cost matrix (e.g. control penalty) associated with the control input (Chui and Chen, 1989).

Alternately, for the controller governing the speed of the USV, separate trajectories must be followed for each phase of the recovery maneuver. As a result, two separate control laws must be derived, although the same procedure is used in the derivation of both control laws. The details of the derivation are shown here only for the waiting phase speed controller; the same derivation applies to the speed controller, except for the desired trajectory the controller attempts to achieve. Only the waiting phase is shown since the expression for the desired trajectory is simpler for the waiting phase than for the entry phase, and as a result the derivation easier to follow.

Equation 6.2, which gives the system of differential equations governing the forward velocity of the USV, shows that the system, while fairly simple, is in strict-feedback form and therefore is well suited for the application of backstepping. Therefore, the x-subsystem is examined first, with the aim of an expression for $u(t)$ which stabilizes

the x-subsystem. This can be done by defining the Lyapunov function:

$$L(t) = (x_d(t) - x_{USV}(t))^2 = (Vt - d - x_{USV}(t))^2 \tag{6.14}$$

By Lyapunov's stability theorem, it is known that since $L(t)$ is positive definite, the system as defined will be stable provided that the derivative of $L(t)$ is negative semi-definite (Lyapunov, 1892), so that:

$$\frac{\partial L}{\partial t} = 2\left(Vt - d - x_{USV}(t)\right)\left(V - u_{USV}(t)\right) < 0 \tag{6.15}$$

where u_{USV} is the velocity of the USV. Applying backstepping theory, this is treated as a control input for the next subsystem in the cascade. Thus, the next step is the determination of a desired velocity which will ensure Equation 6.15 is satisfied (Rugh, 1993). This is made true by the choice of:

$$u_d(t) = V + \gamma_x(Vt - x_{USV}(t) - d) \tag{6.16}$$

for the desired velocity where γ_x is the controller gain. Having obtained a desired expression for u, the problem reduces to determining a control law governing thrust which drives the actual velocity to the desired velocity. Such a control law is derived via Model Reference Adaptive Control (MRAC). The advantage to such an approach is that it allows the actual control law to adapt, based upon the actual observed behavior of the USV. For further details regarding MRAC, see Ioannou and Sun (2012).

The first step in deriving a control law using MRAC is the derivation of a control law for a reference system. Such a system need not be complicated, and in the present

case this particular reference system was chosen:

$$\dot{u}(t) = -u(t) + r(t) \tag{6.17}$$

where $r(t)$ represents the control input to the reference system. As before with the x-subsystem, a Lyapunov function for the reference system is defined, specifically:

$$L_u = (u(t) - u_d(t))^2 = (u(t) - V + \gamma_x (Vt - x_{USV}(t) - d))^2 \tag{6.18}$$

Taking the derivative of this expression with respect to time shows that the reference system is stabilized about the desired trajectory by the choice of:

$$r(t) = V(\gamma_x + \gamma_u) - (Vt - L - x_{USV}(t))\gamma_x\gamma_u - (\gamma_x + \gamma_u - 1)u(t) \tag{6.19}$$

for the control law of the reference system. Here too, terms of γ are controller gains.

Having derived a control law for the reference system, the last remaining step is the derivation of a control law for the true nonlinear u-subsystem. Based upon existing literature, the form of the MRAC control law for a system of the form of Equation 6.2 is (Ioannou & Sun, 2012) is already known to be:

$$T(t) = k_1(t)(au(t) + b) + k_2(t)u(t) + k_3(t)r(t) \tag{6.20}$$

where k_{1-3} represent adaptive gains assigned to the term $(au(t) + b)$ in Equation 6.2, the state variable u, and the reference control law r. These gains are in turn governed

by the differential equations:

$$\begin{aligned}\dot{k}_1(t) &= -\gamma_1 \left(au(t) + b\right)\varepsilon \\ \dot{k}_2(t) &= -\gamma_2 u(t)\varepsilon \\ \dot{k}_3(t) &= -\gamma_3 r(t)\varepsilon\end{aligned} \tag{6.21}$$

with ε representing the tracking error between the true velocity of the USV and the desired velocity, so that:

$$\varepsilon = u(t) - u_d(t) = u(t) - V + \gamma_x \left(Vt - x_{USV}(t) - d\right) \tag{6.22}$$

Thus, by solving the reference system and the differential equations governing the adaptive gains simultaneously with the evolution of the physical system, the controller is able to stabilize the trajectory of the USV about the desired trajectory, while adapting to any changes that may occur in the system.

6.4 Predictor Methods

With a controller defined that is capable of delivering the USV to the recovery vessel at a particular point in time, the choice of when the USV is to enter into the recovery vessel must be addressed. For the stern ramp recovery focused upon here, the goal is ensuring that the USV enters the recovery vessel when the recovery vessel is at a zero pitch angle with a sufficiently small roll angle. Under calm conditions, this may be easily achievable; at higher sea states, this may be the case only periodically, and for very short periods of time. Since the controller and USV require a finite amount of time to execute the entry maneuver, the occurrence of such optimal conditions for entry must be predicted sufficiently far in advance to be of use.

It should be noted that while in this work the focus was upon predicting a zero in

the pitch angle that occurs simultaneously with a sufficiently small roll angle, the same overall methodology could be applied in a scenario with different positional requirements. For example, in some circumstances performing the recovery operation at a zero roll angle may be more important than the pitch angle. Additionally, each predicted behavior was made with a separate prediction; meaning, to predict roll and pitch for a given data set there would be two separate predictors (e.g. neural networks), the outputs of those predictors would then be compared to find like times of zero pitch angle and a sufficiently small roll angle. Thus, the predictor serves as the switch governing when the controller transitions from the station keeping mode to the entry mode. This is visualized graphically in Figure 6.4. Hence, the ship motion prediction algorithm is truly an integrated portion of the controller.

6.4.1 Neural Network Prediction

The first prediction algorithm considered for use with the controller was an artificial neural network. Gougoulidis (2008) explained the advantages of artificial neural networks for marine applications. He specifically discusses ship design, but his reasons apply to recovery operations as well. First, neural networks have a parallel structure which increases the ability to deal with complex problems. They also have great predictive capabilities where little information exists about the inputs and outputs of a problem. Lastly, they excel at dealing with highly nonlinear problems (Gougoulidis, 2008). For some examples of past successful application of neural networks for multiple classes of ship motion problems see Haddara and Xu, 1999; Hess et al, 2006a,b; Lainiotis et al, 1993; Xing and McCue, 2009, 2010.

The neural network used in this study was an open loop neural network trained using previous state data (from the previously discussed DTMB model) representing the behavior of the recovery vessel. It should be noted that while much of the aforemen-

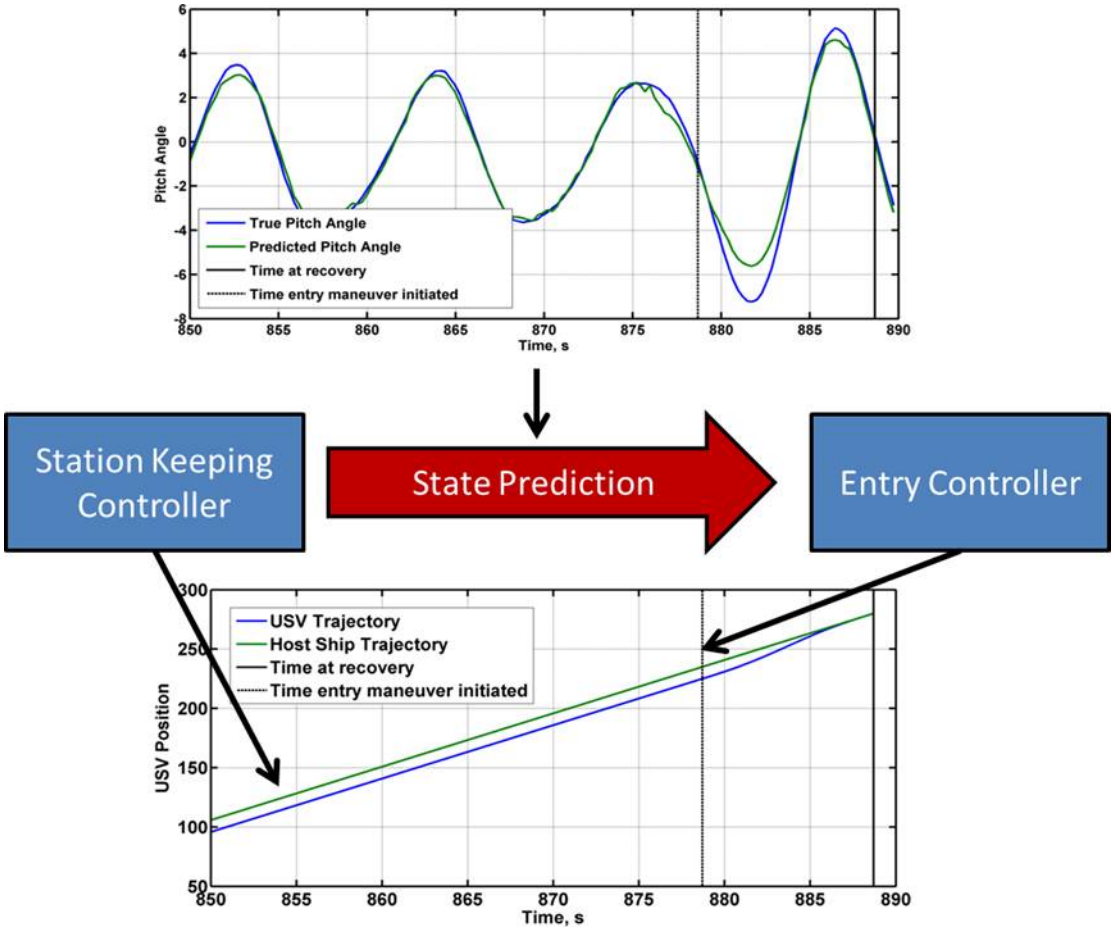


Figure 6.4: Example of the operation of the combined nearest neighbor prediction algorithm and control system. The controller operated in station keeping mode until the neural network predicts conditions desirable for execution of the entry maneuver. This results in the controller switching from the station keeping mode to the entry mode at the appropriate time. Note that the first vertical line in the figures indicates the time at which the entry maneuver begins; the second indicates the end of the entry maneuver, when the USV has successfully been recovered by the host ship.

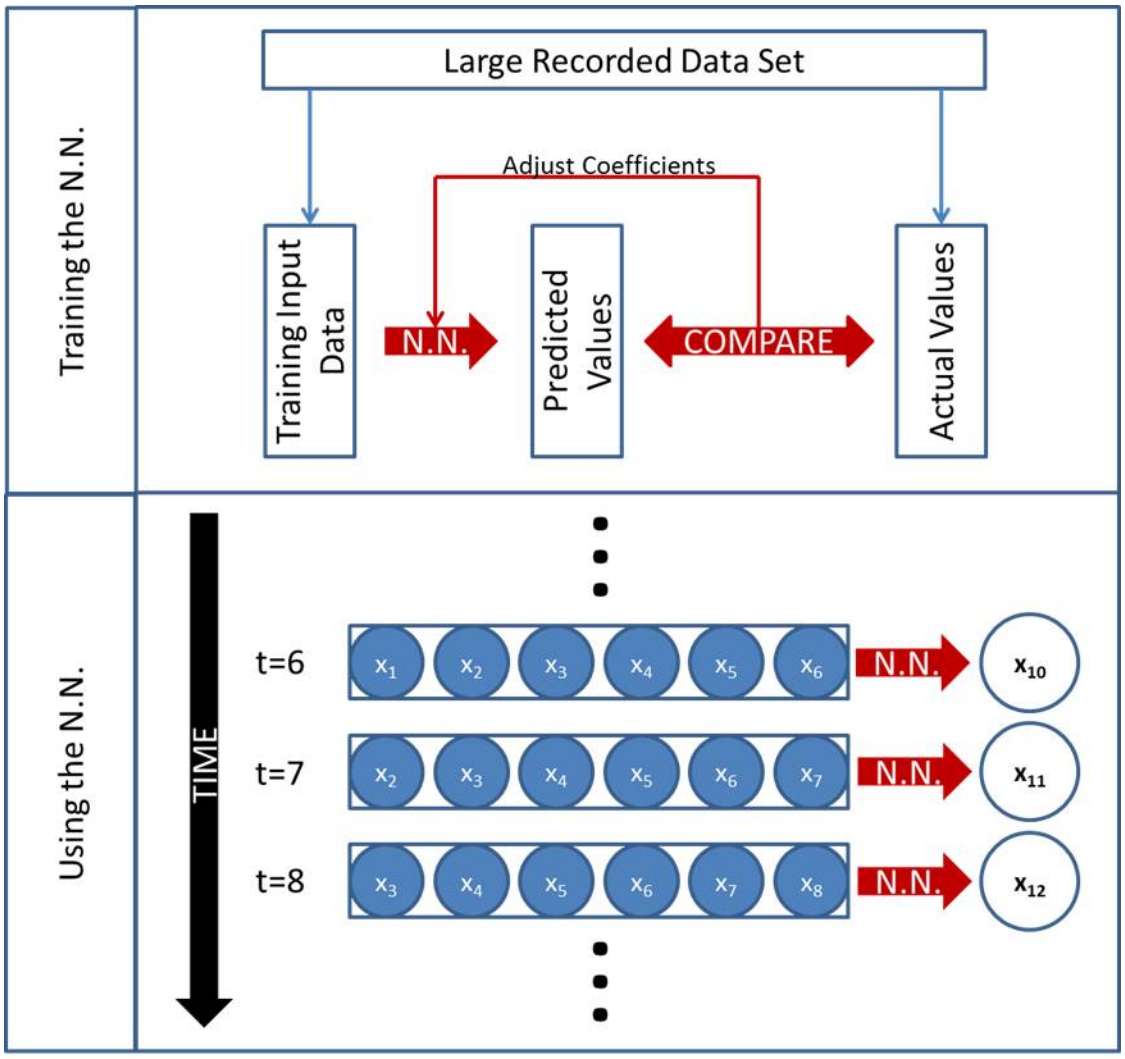


Figure 6.5: Training and evaluation of the neural networks utilized in the present study. During training (top), input data segments are extracted from large sets of recorded data. The corresponding future values are also extracted, and the neural network coefficient values are iteratively solved so that the values predicted by the neural network in the training data match the actual extracted values. Meanwhile, when being evaluated (bottom), all data points recorded in the immediate past (for example, the past five seconds) are supplied to the neural network as an input. The neural network then provides a prediction of the future ship state at the predefined distance into the future. X represents a state vector of the time series, with indices corresponding to specific instances in time for the time series. Solid circles represent actual values, while empty circles correspond to predictions of values.

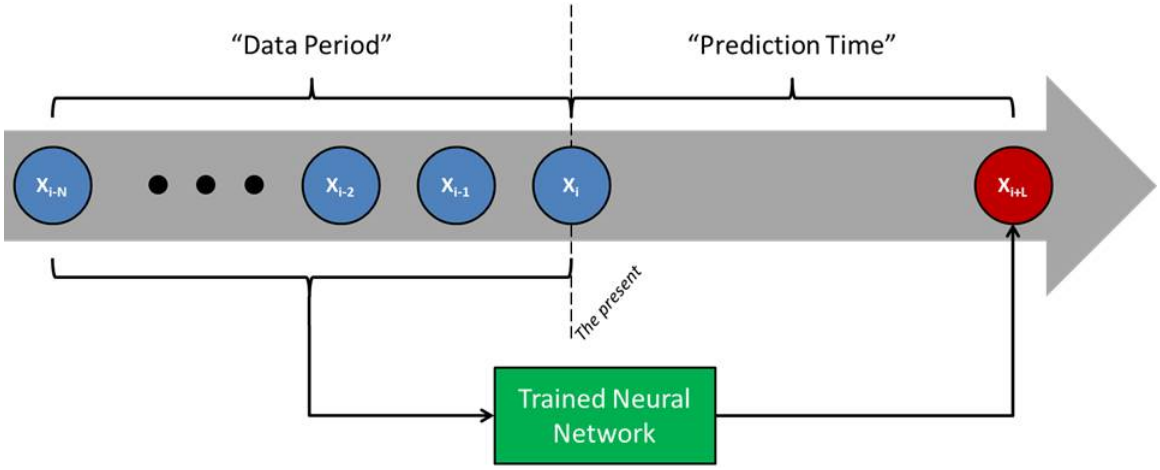


Figure 6.6: Behavior of the neural network. The most recent data recorded over an amount of time denoted as the "data period" is used to predict some previously defined time into the future. This amount of time is denoted the "prediction time" of the neural network.

tioned data set is concerned with capsize, the data set also includes a fair number of seakeeping runs where capsize did not occur. A pictorial visualization of the training and evaluation of such a network is shown in Figure 6.5 where $\mathbf{x}$ denotes a state vector with subscripts indicating different instances in time. As opposed to a closed loop network, where predictions made by the network can be fed back into the network as inputs (so that predictions farther into the future can be made in an iterative manner), the open loop network directly predicts the state of the recovering vessel at some future time, henceforth referred to as prediction time. The input data for making such a prediction is a time history of the vessels behavior starting at some point of time in the past and extending to the present; this period of time is henceforth referred to as the data period. A schematic demonstrating the operation of such a network is shown in Figure 6.6. The total number of internal (hidden) layers comprising the network was based upon the amount of complexity necessary so as to achieve acceptable results during the training of the network. In this case a 15 layer neural network was utilized.

One key point requiring discussion was that the neural network was not used to predict pitch and roll angles directly. Rather, it was instead used to predict a function whose value spikes at times when the angle is zero. There are two steps in formulating this function. The first is to establish a function $f(t)$ that defines the time until the next zero value which is further into the future than the time required to execute the entry maneuver. For example, if at a given point in time, the next zero pitch angle occurs in ten seconds, and the entry maneuver only required eight seconds, the value of $f(t)$ at that point in time would be ten seconds. However, if the entry maneuver time was thirteen seconds, the value of $f(t)$ would be based upon the time to the next zero (not the one occurring in ten seconds, since this zero does not allow enough time for the recovery maneuver to be executed). Note that when training the neural network, the entire data set is available, so the determination of values for $f(t)$ is straightforward. Once $f(t)$ has been determined, the actual target function can be evaluated. This target function $g(t)$ is defined as:

$$g(t) = \frac{1}{f(t)} \tag{6.23}$$

The form of this function is advantageous, since it spikes to a value of n, that is built into $f(t)$, at times when a zero occurs and remains comparatively small at all other times. This function is the output predicted by the neural network; local maxima are found and then evaluated to determine whether the value is high enough to classify it as a zero roll. Using $g(t)$ as the output of the neural network was found to allow zeros to be predicted with substantially less errors than in the case when zeros are predicted directly from a prediction of the pitch and roll angles, respectively.

These open loop neural networks were implemented in MATLAB (MathWorks, Inc, 2013) as nonlinear autoregressive neural networks. Details regarding the neural net-

work design can be found in the MATLAB Neural Network Toolbox User Manual (2013). Such neural networks are built into the MATLABs neural network toolbox, and can be trained on time series data using built-in MATLAB commands, making for convenient implementation. A subset of the data was used to pre-train a neural network. That neural network was then fed inputs of vessel state over a finite amount of previous time. This period of previous data upon which a prediction is based is referred to as the data period as per Figure 6.6. This data is used by the trained neural network to provide a prediction of state at a single time instance in the future; this time period is referred to as the prediction time. The process is incrementally repeated as shown in Figure 6.5. To clarify, a given choice of prediction time and data period results in a single neural network, trained as described previously, which can be used to predict the motion of the recovery ship under the subset of conditions represented by the training data. Network training was performed on a computer with 64 AMD Opteron series 6276 (2.3 GHz) processors with 2 GB of RAM per core. Each network was trained on a separate core. In general, the training calculations took between 60 - 90 minutes, depending upon the supplied data period and prediction time of the network.

For the purposes of the present work, the recovery vessel is the notional destroyer DTMB 5514 hull form, due to the existence and availability of experimental seakeeping data for this hull form. Specifically, this data was taken from a series of seakeeping and dynamic stability experiments performed at the Carderock Division of the Naval Surface Warfare Center in 2006 (Hayden et al, 2006). These experiments were performed in a wave basin on a 1/46.6th model. Each neural network produced was trained using 152 separate experiments, while a different set of 20 experiments was used to evaluate the neural network performance. While the length of the experimental data sets varied, a typical acquisition corresponded to 7-9 minutes of full-scale

time, representing anywhere from 36 to 64 individual wave cycles.

6.4.2 Nearest Neighbor Prediction

The second ship motion prediction algorithm considered is based upon searching a database of historical ship motion data for data points very similar to the current conditions (McCue 2008). Once suitable neighboring points matching the current state are identified in the historical data, motion prediction can be made by identifying the conditions observed at an appropriate later time in the historical data. For example, to make a prediction ten seconds into the future, one would identify the state ten seconds after each identified neighbor point in the historical data. In prior work by McCue, the state vector used for comparison consists of the roll angle and velocity and the pitch angle and velocity (McCue 2008).

In this work, the state vector was expanded to include heave as well. Thus, the state vector used in this work consisted of the roll angle, the roll velocity, the pitch angle, the pitch velocity, and heave. It is also important to note that each component of the state vector is normalized by its standard deviation. The first step in this method is identifying the points where the difference between the current state and the prior state is at a minimum . This difference is defined as using the 2-norm as:

$$\ell(t) = |\overrightarrow{s}(t) - \overrightarrow{s}(t_0)|\,, \qquad t < t_0 \tag{6.24}$$

Note that here $\overrightarrow{s}(t)$ represents the total state of the host ship at time t. A set of times t_i where this distances reaches a minimum is used to generate the prediction of the future state at some future time t_1. An individual estimate of the future state is generated from each of these neighbor points; the mean of these individual predictions being used to generate the final prediction. More formally, the predicted future state

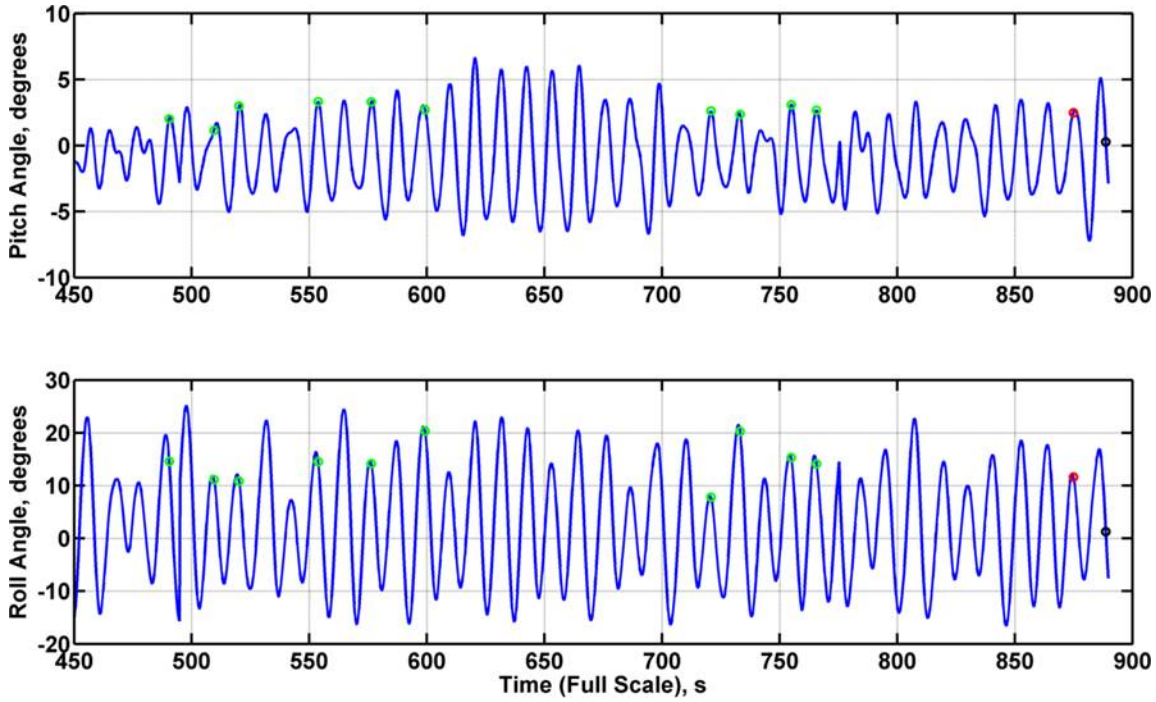


Figure 6.7: Operation of the nearest neighbor predictor. The red dot indicates the state when the prediction of future state was made. The black dot indicates the predicted future state, while the green dots indicate the nearest neighbors found in the data acquired prior to the time at which the prediction was made. In this example, the actual pitch angle at the time when a zero pitch angle was predicted is 0.17 degrees, while the roll angle at the same time is 3.05 degrees.

can be defined as:

$$\overrightarrow{s}_{est}(t_1) = \frac{\sum_{i=1}^{N} \overrightarrow{s}\,(t_i + \Delta t)}{N} \tag{6.25}$$

where in this study N was set equal to 10. A demonstration of the operation of this predictor is shown in Figure 6.7.

The prediction approach was evaluated using the same DTMB 5514 data set used to train and evaluate the neural network data. Each experimental data set was considered individually, with historical data coming only from previous portions of that file. The individual predictions obtained from the ten neighbor points identified in

the historical data set identified as being the closest to the current data point were averaged to form a prediction of the future state of the system.

For the nearest neighbor and the neural network, the emphasis of testing was to determine how accurately the prediction algorithms were able to predict zeros in roll and pitch behavior. Specifically, the calculated errors, ε, represent how large the actual angle was when angle was predicted, so that:

$$\varepsilon = \theta_{actual}(t_1), \qquad \theta_{est}(t_1) = 0 \tag{6.26}$$

To gain additional insight into the performance of the prediction algorithms, a normalized error metric was considered, which was defined as:

$$\varepsilon^* = \frac{\theta_{actual}(t_1)}{\sigma} \tag{6.27}$$

where σ represents the standard deviation of the actual angle (either pitch or roll).

6.5 Results and Discussion

6.5.1 Controller Results

Numerical simulations were performed for the execution of the recovery maneuver under a variety of conditions. For these results, the USV is assumed to be the RibCraft USV studied by Sonnenburg (Sonnenburg, 2012), while the recovery vessel is assumed to be the DTMB Model 5514 hull form (Hayden, 2006). The relevant parameter values for this USV are shown in Table 6.1, while the recovery vessel moved on a constant course. Figure 6.8 considers a recovery vessel forward velocity of 9 knots. Different curves within this figure demonstrate the effects of various trailing distances and maneuver times. Figure 6.9 shows results for a recovery ship moving with a forward

Parameter	Symbol	Value
Yaw Rate Time Constant	τ	0.65
Rudder Coefficient	K	0.33
Quadratic Drag Coefficient	a	0.1324
Linear Drag Coefficient	b	0

Table 6.1: Model parameter values for the Ribcraft USV used in simulations (Sonnenburg, 2012)

velocity of 18 knots, with the same variation of trailing distances and maneuver times as in Figure 6.8.

Overall, the controller exhibits very good performance, but it is important to note that performance is limited by the amount of time that must be allocated for the entry maneuver. If insufficient time is allocated, the USV is unable to decelerate back down to the speed of the recovery ship without applying reverse thrust. For reasons of simplicity, the mathematical model does not consider reverse thrust. Although some vessels may be able to smoothly switch from forward to reverse thrust, for the purposes of this work this was considered unacceptable. Hence, the existence of reverse thrust indicates a scenario from which successful recovery is likely not practically possible. Rather, the predictor algorithm is required to provide sufficient warning time of the desired conditions so that the entry maneuver can be executed without reverse thrust.

Another valid concern regarding the design of the controller is its robustness to unmodeled dynamics, such as those induced by the sea state. To investigate this potential issue, simulations similar to those performed in Figure 6.8 (a recovery vessel surge velocity of 9 knots), are shown in Figure 10, but with the addition of unmodeled wave forcing acting in surge. This forcing was sinusoidal, and had constant amplitude (within a given simulation) and random phase. Multiple wave frequencies were considered. This acted as an unknown force term in the surge equations of motion

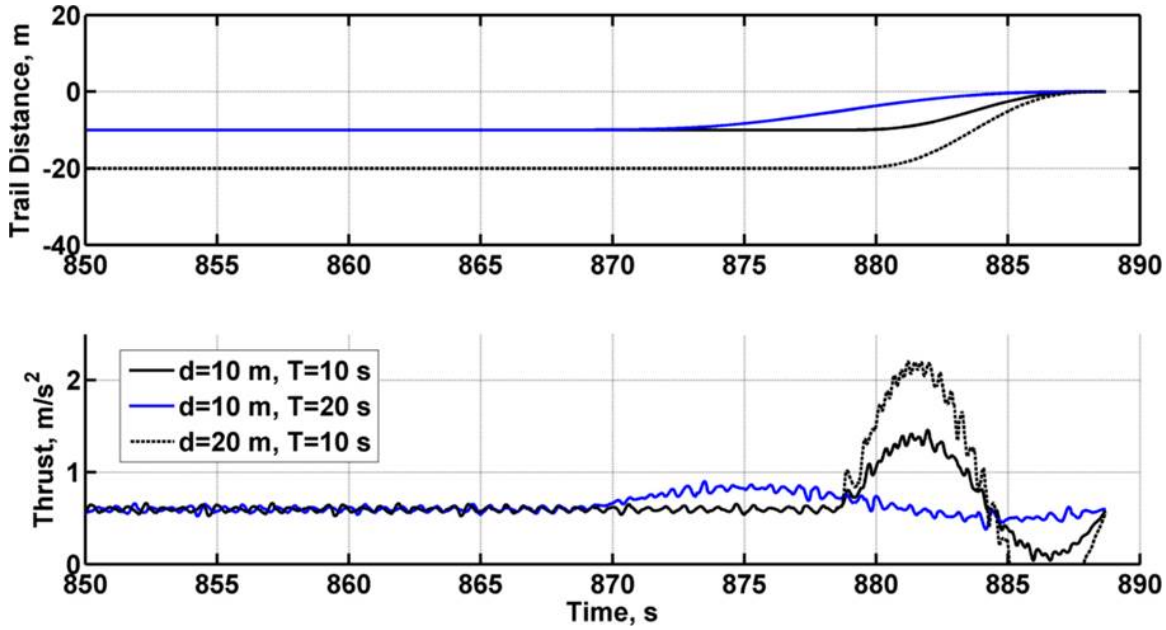


Figure 6.8: Surge position relative to the recovery ship (top) and acceleration (bottom) during the waiting and entry phases of the recovery maneuver, executed at 9 knots. The solid black line corresponds to a trailing length of 10 m and a maneuver time of 10 s. The solid blue line corresponds to a trailing length of 10 m and a maneuver time of 20 s. The dashed line corresponds to a trailing length of 20 m and a maneuver time of 10 s. Note that the final case (with a 20 m trailing length) is not physically viable without seamless thrust reversal.

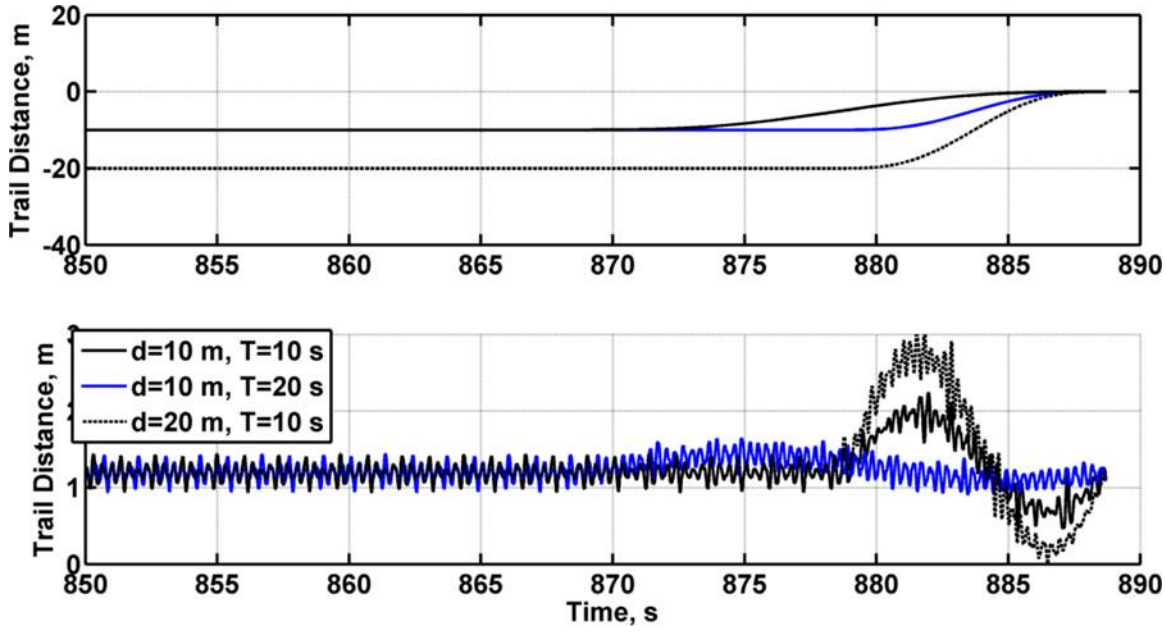


Figure 6.9: Surge position (top) and acceleration (bottom) during the waiting and entry phases of the recovery maneuver, executed at 18 knots. The solid black line corresponds to a trailing length of 10 m and a maneuver time of 10 s. The solid blue line corresponds to a trailing length of 10 m and a maneuver time of 20 s. The dashed black line corresponds to a trailing length of 20 m and a maneuver time of 10 s.

(Equation 6.2), one that was unknown to the controller. Simulations were performed at 20 different amplitude levels, ranging from 0 to 1 g (note that these accelerations represent forces scaled by the mass of the USV). At the end of these simulations (e.g. the time at which the USV is to be recovered) a residual was calculated, which represents the distance between the actual and intended positions of the USV at the time when the entry maneuver is intended to be completed (and which corresponds to the time t_0 at which zero pitch angle was predicted):

$$R = \|x(t_0) - x_d(t_0)\| \tag{6.28}$$

Twenty individual simulations were performed at each individual value of the wave amplitude. The mean results (averaged over wave phase) from these simulations are shown in Figure 6.10. Over these cases, it appears that for the defined system, the controller is robust to unmodeled phenomena in the wave state.

As shown by Figures 6.8 and 6.9, at increased recovery vessel velocities, the maneuver can be performed in less time. This is due to the nonlinearity of the drag acting upon the USV; at higher velocities, the USV is able to decelerate back down to the velocity of the recovery vessel more rapidly than at lower velocities. The minimum maneuver time possible without the application of reverse thrust is shown in Figure 6.11 as a function of recovery ship velocity. From these results, it can be concluded that approximately 8 to 18 seconds are required for this sample USV to successfully execute the entry phase of the maneuver.

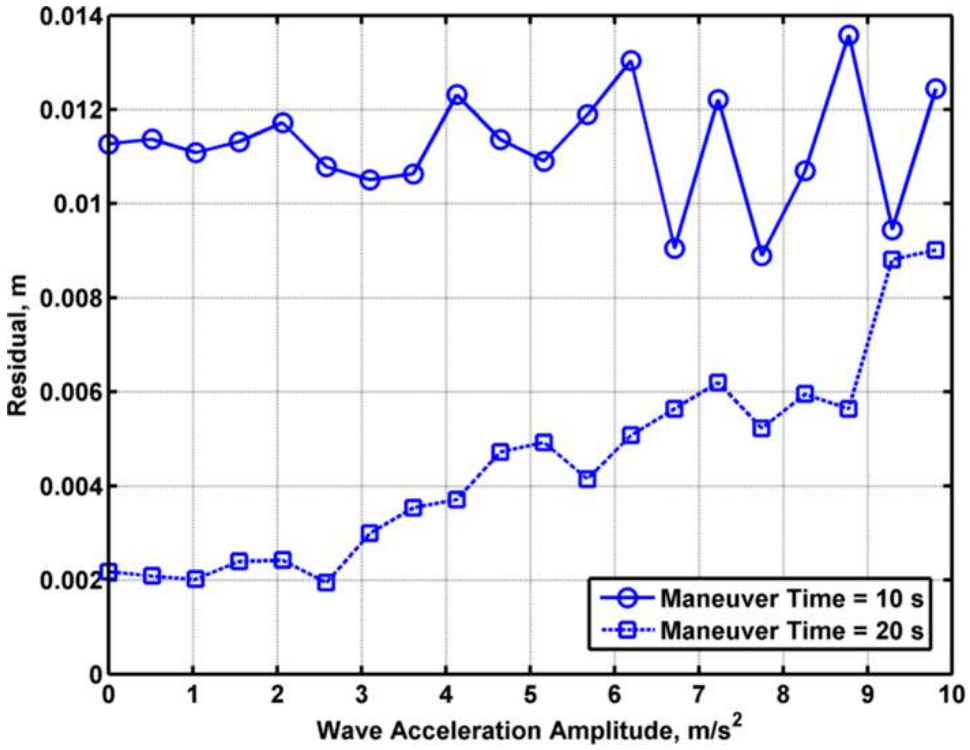


Figure 6.10: Results from the controller robustness study. The residual represents the distance between the actual and intended position of the USV at the moment it is to be captured. These results indicate that the residuals resulting from the unmodeled sea state are sufficiently small that the sea state can remain unmodeled for the studied recovery paradigms.

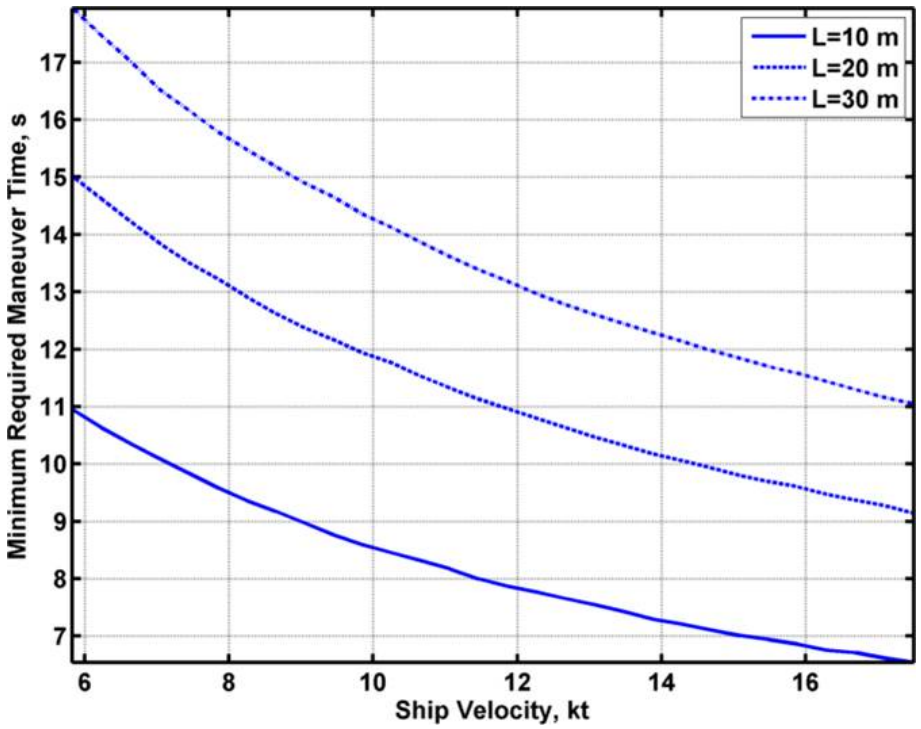


Figure 6.11: Minimum maneuver time which must be allotted for the entry phase of the recovery maneuver, as a function of the velocity of the recovery ship and the trailing distance from which the USV begins the maneuver.

6.5.2 Predictor Results

The approximately 8 - 18 seconds required to execute the entry maneuver represents a minimum requirement for the prediction time required for the controller to operate successfully. The predictor algorithm must predict favorable conditions sufficiently far in advance that the controller can execute the maneuver without reverse thrust. It is in the light of this requirement that the performance of the previously discussed neural network and nearest neighbor algorithms must be evaluated.

As discussed previously, the performance of the neural network was evaluated using 20 experiments separate from the experimental data used for training. During the training, it was found that neural networks given multiple degrees of freedom performed only marginally better than those given only the desired predicted behavior data and increased computation time significantly. As result, neural networks were computed for roll behavior using roll data as inputs to the neural network and the same was done for pitch. In numerical experiments, the neural network described previously was found to reliably predict zeros in the roll and pitch behavior eight seconds into the future. At this level of prediction, the normalized median errors in pitch and roll are 0.14 and 0.16, corresponding to absolute errors of 0.88 and 2.7 degrees in pitch and roll, respectively (note that most of the data set had significantly larger roll motions). While this meets the minimum requirements of the controller, there are several methods through which the performance of the network might be improved. For instance, information regarding the upcoming wave train might be acquired via forward looking radar or some other mechanism, increasing the amount of information available to the neural network.

The second predictor tested, the nearest neighbor algorithm, was found to estimate zeros very accurately. At approximately 28 seconds of full scale time, the algorithm

was able to successfully predict zeros in pitch, with a normalized median error of 0.047. This corresponds to less than 0.34 degrees of error, which is quite sufficient for the purposes of the controller. Similar accuracy was obtained with regard to the prediction of roll motions, with a normalized median error of 0.08, corresponding to 0.62 degrees. One advantage of this algorithm is that the principal errors seem to be related to the amplitude of the motion, rather than its phase. In other words, errors in calculating when a zero will occur are reduced relative to the errors associated with predicting how large a motion will be, which is not necessary in this application. An example of this type of behavior is shown in the upper plot of Figure 6.5.

6.6 Conclusions

In this work, a controller for the automated recovery of USVs via stern ramp, based upon a nonlinear ship dynamics model and nonlinear ship motion prediction algorithms, has been derived and numerically tested. This controller is combined with a predictor algorithm, which predicts the future state of the recovery ship. This predictor serves as a trigger for the controller, so that the controller can ensure that recovery occurs at the desired state of the recovery ship. Two individual prediction algorithms were investigated in this work, one based upon artificial neural networks and another based upon the identification of neighbor points in historical data.

The robust control laws and prediction algorithms developed in this chapter represents a successful proof of concept for the use of ship motion prediction to aid in the successful deployment of recovery controllers. A nonlinear model, combined with nonlinear control, has been used to generate a robust and adaptive controller that appears to meet the challenges of a demanding real world application. The development of such a control law represents the final step in the process that has been the central

theme throughout this dissertation, that of taking nonlinear models, validiting their representation of reality, understanding their behavior, and finally deploying them into meaningful real-world applications.

Future work should focus on experimental validation of the controller and incorporation of seaway data and supplementary state data into the neural network algorithm with the hope of improving its prediction capabilities. Going a step further, one might argue in the future that non-trivial motions could be viable during a recovery process if the host vessel and autonomous vessel motions are matched, e.g. minimizing relative motion between the two vessels. This is a ripe area for future experimental and computational research to determine if indeed safe recoveries while undergoing non-trivial absolute motions with minimal relative motions are feasible.

CHAPTER VII

Conclusions

Over the past 60 years, dramatic increases in computational power have revolutionized the scope and importance of modeling and simulation in science. This has enabled technologies and understanding which would not otherwise have been possible. However, it has also created challenges. Often, the phenomena which require such computational power to investigate are both complex and nonlinear in their dynamics; they are potentially even chaotic. The validation and control of models with such complex dynamics to real-world problems is rarely as straightforward as the application of simpler models. The overarching goal of this dissertation has been to contribute both generally and acutely towards the diminishment of these difficulties. Each chapter contributes to this goal in its own way, as summarized below.

Chapter 2 focuses upon the examination of physics based system response quantities for the validation of chaotic simulations. This work focused on testing a set of metrics on canonical chaotic systems. The advantage of this is that the values calculated for the candidate validation metrics could be compared with the known behavior of the underlying dynamical system. For example, with the Lorenz attractor, it is well understood where bifurcations in the system response occur. As a result, the values of the candidate validation metrics can be investigated to ensure that they identify the

change in dynamics associated with these bifurcations. This would not necessarily have been the case with a more complex numerical system (such as that applied in Chapter 3), making this work an invaluable first step.

The emphasis in this chapter was to ensure that each candidate metric was capable of resolving strong changes in system dynamics, particularly those associated with bifurcations. This work essentially served as a screen for candidate validation metrics prior to undertaking more detailed (and more ship dynamics specific) testing. All of the candidate metrics considered to be worthwhile validation metrics were found to be sensitive in a useful manner (from a validation perspective) to such dramatic changes in the system dynamics, and were carried on to the next study.

While Chapter 2 investigated the utility of the candidate validation metrics using canonical systems, Chapter 3 studied the validity of the candidate validation metrics using a numerical model more aligned with real world validation applications. Specifically, a state of the art commercial nonlinear ship dynamics model was employed, and used to simulate a series of dynamic seakeeping experiments. Simulations were performed under conditions matching over 150 experimental data sets. For each set of conditions (matching an experiment), simulations were performed for 60 separate initial conditions and three separate physics models (the simulation program allows for variation in the complexity of the physics model). This experiment allowed the metrics to be tested under conditions akin to real world applications, and for simulation metric values to be compared with experimental metric values. From this study, it was possible to make conclusions regarding which metrics were best able to separate trajectory changes due to differences in the underlying dynamics from those due to a difference in the initial condition. This allowed recommendations to be made regarding which of the candidate metrics are best suited for use as validation metrics

for simulations of chaotic systems.

In addition to these two studies into appropriate choices for validation metrics for chaotic time series simulations, the third study (Chapter 4) included in this dissertation focused on performing validation experiments in support of efforts to model the dynamics associated with the slamming of compliant bodies. These experiments consisted of a wedge shape formed by a stretched rubber membrane. This structure was driven into water via a pneumatic piston at various velocities. A high speed camera was used to visualize the resulting strains in the rubber membrane forming the wedge shape. This work contributes to the previously mentioned goal by providing the experimental data necessary to enable validation. It is important to remember that setup of the validation experiment should be treated with at least as much care as the simulation and the actual validation analysis – validation experiments cannot be an afterthought in the process.

Chapter 4 rounded out the validation portion of this dissertation. The next chapter focused upon the use of Non-Intrusive Polynomial Chaos (NIPC) to more efficiently investigate the statistical distributions of random processes. This method has extremely wide applications, but the foremost application is likely quantification of uncertainty. In this study, NIPC was applied to numerical simulations of parametric roll, which is highly nonlinear (and since the APL China incident in 1998, highly relevant) ship dynamics phenomena, which can result in extremely high roll angles and capsize. In this study, ship speed, wave frequency and wave angle were treated as random inputs with known input distributions. The maximum roll angle observed in a simulation represented a random process dependent upon the known input distributions. Thus, in this example scenario, NIPC was used to generate a probability distribution for the maximum roll conditions based upon an input distributions for

the ship and sea conditions. Due to the complexity of the dynamics underpinning these simulations, this represents a very challenging application of NIPC.

A sufficiently large number of simulations (1500) were performed so that NIPC could be compared with results from the Monte Carlo method. Full statistical distributions for both NIPC and Monte Carlo were compared, as were statistical moments calculated using each method. Under most circumstances, NIPC was found to produce results very similar to those generated by the Monte Carlo method, but with substantially increased computational efficiency. Comparable results were obtained using approximately thirty simulations with NIPC in contrast to 1500 simulations with Monte Carlo. The tradeoff here was found to be that NIPC had substantially decreased computational efficiency when there is a substantial probability of capsize, to the extent that the Monte Carlo approach may well be less computationally expensive. Nevertheless, when this restriction is satisfied, NIPC represents an excellent method for minimizing the number of computationally expensive simulations required to extract statistically meaningful information from a nonlinear dynamics model under stochastic conditions.

The final topic, found in Chapter 6, considered in this dissertation is the development of control algorithms for automating the recovery of USVs in elevated sea states. Here, a nonlinear model of the USV dynamics was used to develop a controller for the USV based upon the combination of backstepping and Model Reference Adaptive Control (MRAC). The most critical challenge in this work was that in elevated sea states, recovery can only occur when the recovery ship is at small roll and pitch angles. As a result, there are only very small windows of time during which the actual entry of the USV into the recovery vessel can be executed. A key feature of this controller is its use of ship motion prediction algorithms to predict the use of such windows, with

the terminal phase of the USVs recovery maneuver being initiated so that the USV enters the recovery vessel during such a window.

A key component of this work was the determination of the minimum time in the future an entry window (with low roll and pitch angles) must be predicted for a recovery maneuver to be executed. Two separate motion prediction algorithms were tested against this criteria. The first of these involved an artificial neural network, while the second involved the application of a nearest neighbors approach to historical data for the recovery ship. Both predictive algorithms, but the nearest neighbor approach, were found to meet the needs of the controller. The robustness of the controller was investigated in numerical experiments, where it was found to be very robust against such phenomena. Ultimately, the controller was deemed a viable approach for automating the recovery of USVs at high sea states, one worthy of experimental testing.

Such an experimental investigation into the use of the derived controller represents the most logical future investigation building upon the control work. The other areas of investigation also provide opportunities for future work. For the validation work comprising Chapter 2 and Chapter 3 of this dissertation, there are two main avenues of future investigation. The first of these would be to expand the body of nonlinear system metrics test. While careful thought went into selecting the metrics used in this study, they certainly do not represent a comprehensive list of viable metrics and it is entirely possible that other metrics which would add valuable insight when used alongside the metrics presented here. This work has provided the framework for such investigation into additional candidate validation metrics. The other main path forward is putting the recommended metrics into practice validating meaningful, real world simulations.

Regarding the NIPC work, the most impactful future investigation from the perspective of the author would involve the analysis of errors and uncertainty in the NIPC method. One of the clear advantages of the Monte Carlo method is its inherent simplicity – it is easy to understand how measurement or simulation errors affecting the input data for a Monte Carlo analysis might affect the end result. This is substantially less clear with the NIPC approach. Such an investigation would allow a better characterization of when an user is best served by taking advantage of the potential computational savings of the NIPC approach and when one ought to appeal to the rigorous nature of the Monte Carlo method. While this is understood in general (NIPC is most beneficial when the physics are limited in order and data points are expensive to generate), a study such as that proposed would allow for a clear demarcation of the regime within which the use of NIPC is justified and worthwhile.

As discussed previously, one important path forward with the USV recovery controller work involves taking the controller into the real world via experimental investigation. However, this is not the only area worthy of consideration. Ship motion prediction algorithms have applications far beyond the recovery of USVs; improvements in the state of this art would represent a substantial contribution. It should also be pointed out that only a single recovery method (stern recovery) was considered in this work. Several different recovery methods exist and are in use and it may prove beneficial to extend the framework laid out in this study to those recovery methods as well.

Taken together, these works represents an amalgam of contributions, each of which contributes to the overall goal of enabling the use of complex, nonlinear, and chaotic numerical models for real world applications. Each piece aids a different stage of this process. The author believes that this body of work contributes to the community by filling in gaps in the existing body of literature and bridging obstacles between

numerical models of ship dynamics and real-world applications.

APPENDICES

APPENDIX A

Phase Space Plots

This appendix presents additional phase portraits analogous to those presented for Run 213 in Chapter 3. Accordingly, phase portraits are presented for two additional runs, Run 296 and Run 409. Such phase portraits are plotted for each code version, in keeping with the presentation of Run 213 in Chapter 3. Phase portraits for Run 296 are shown in Figure A.1, Figure A.2, Figure A.3, and Figure A.4. Phase portraits for Run 409 are shown in Figure A.5, Figure A.6, Figure A.7, and Figure A.8.

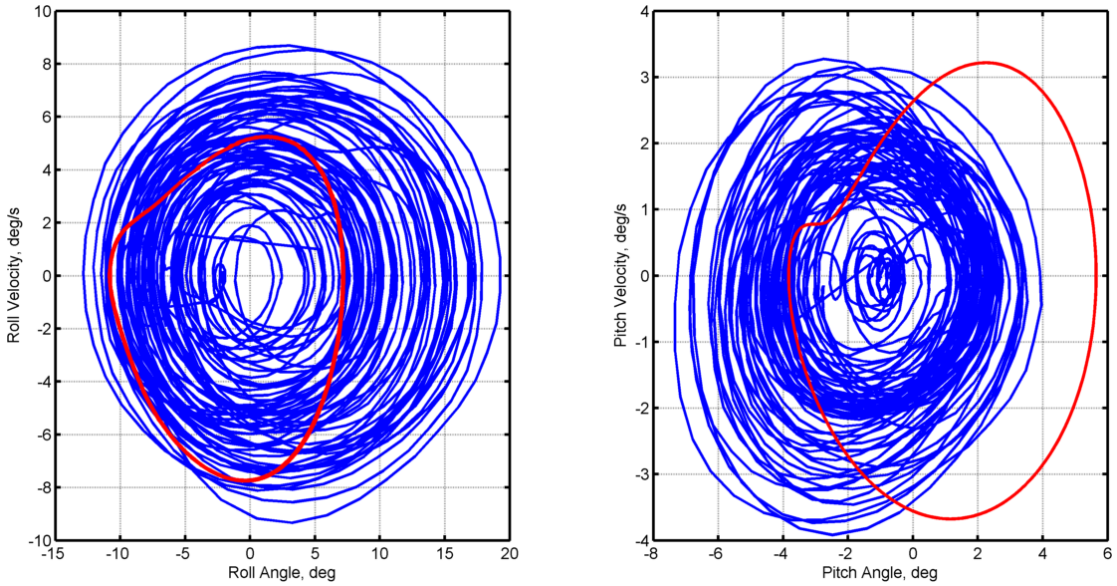


Figure A.1: Comparison of experimental phase portraits with the phase portraits from the linear simulation. Red lines correspond to simulated data; blue corresponds to experimental data. Data for experimental conditions matching Run 296.

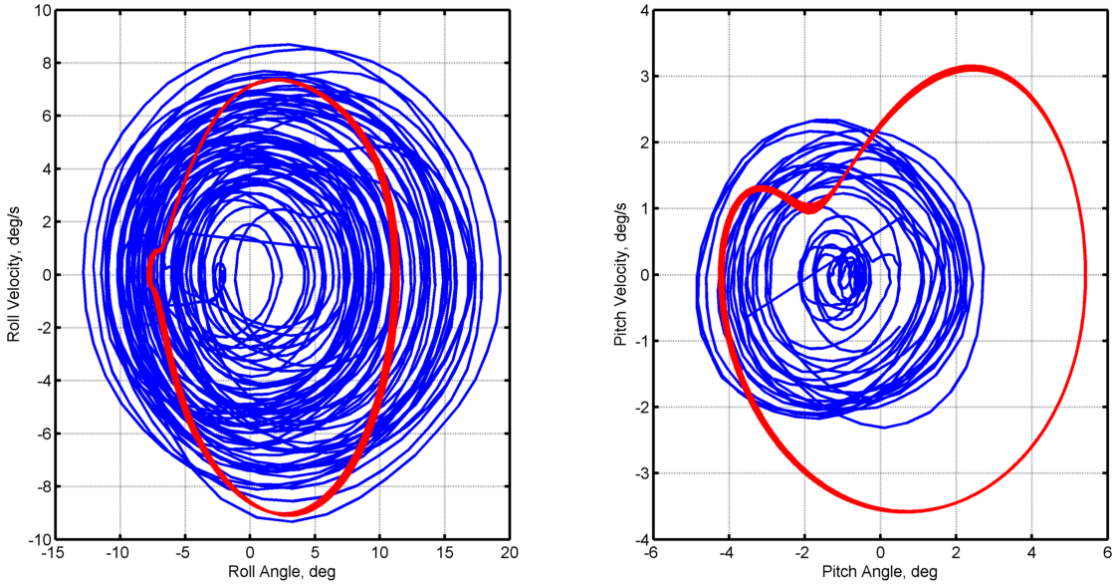


Figure A.2: Comparison of experimental phase portraits with the phase portraits from the approximate-small simulation (approximately linear with small lateral motions). Red lines correspond to simulated data; blue corresponds to experimental data. Data for experimental conditions matching Run 296.

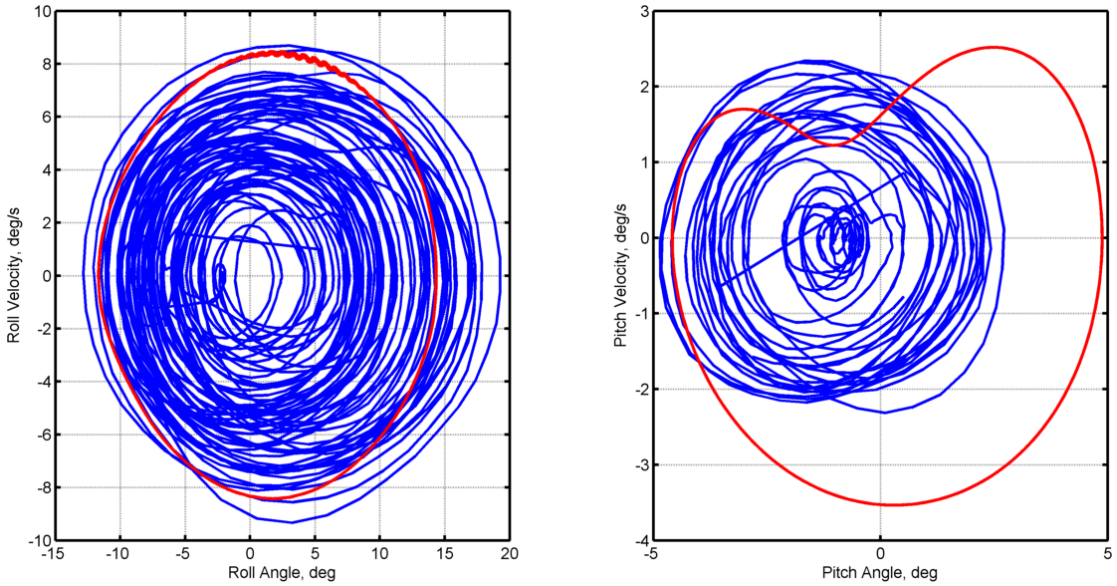


Figure A.3: Comparison of experimental phase portraits with the phase portraits from the approximate-large simulation (approximately linear with large lateral motions). Red lines correspond to simulated data; blue corresponds to experimental data. Data for experimental conditions matching Run 296.

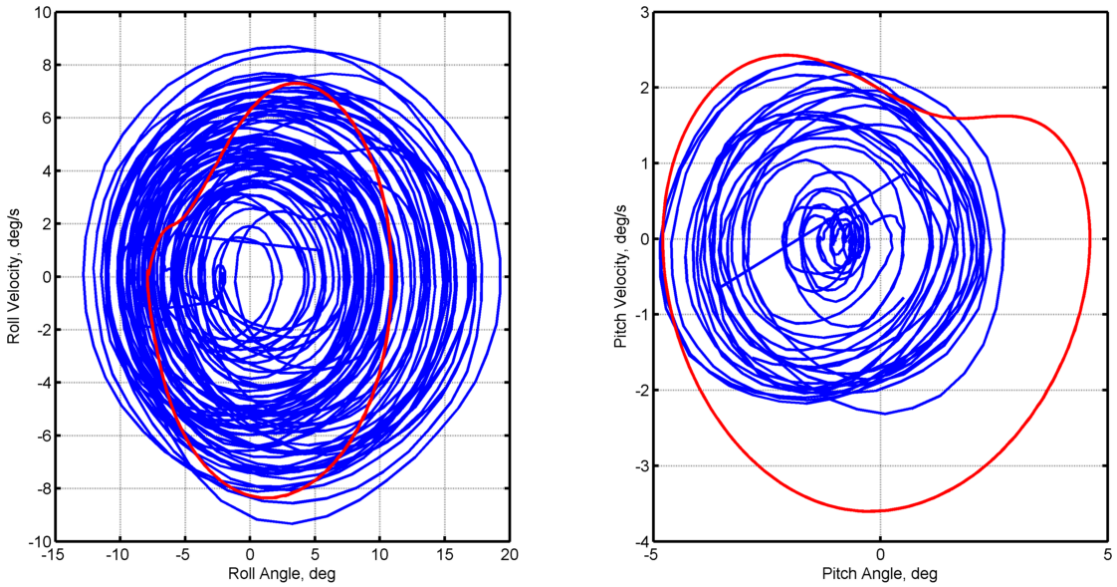


Figure A.4: Comparison of experimental phase portraits with the phase portraits from the fully nonlinear simulation. Red lines correspond to simulated data; blue corresponds to experimental data. Data for experimental conditions matching Run 296.

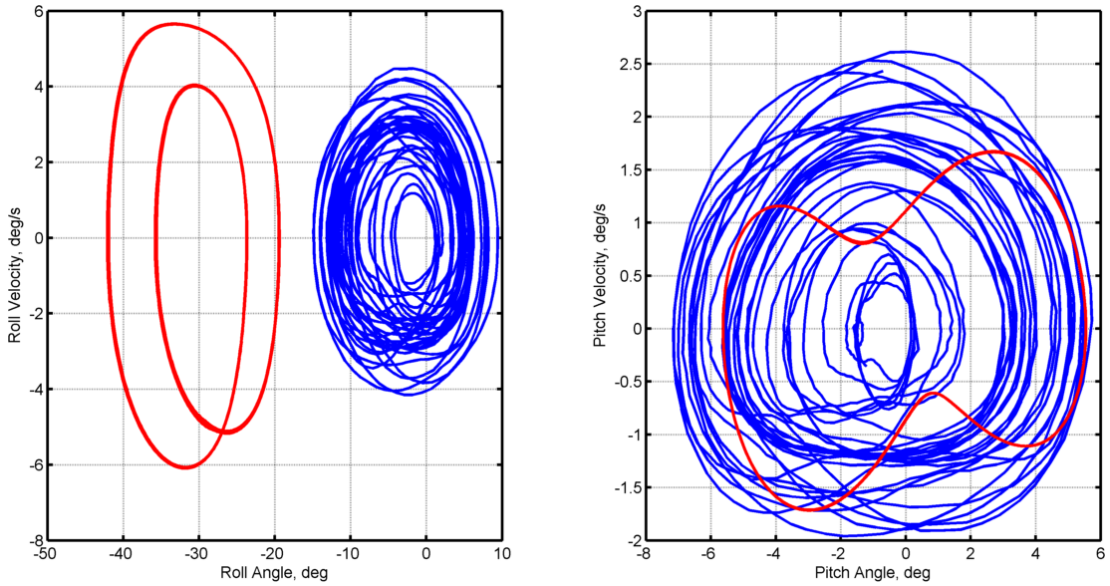


Figure A.5: Comparison of experimental phase portraits with the phase portraits from the linear simulation. Red lines correspond to simulated data; blue corresponds to experimental data. Data for experimental conditions matching Run 409.

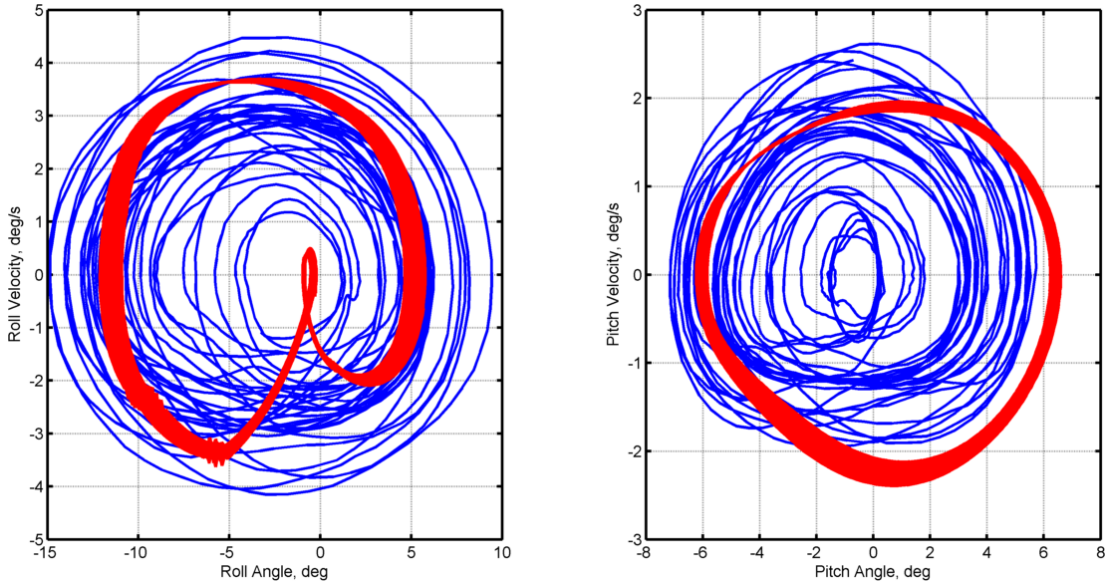


Figure A.6: Comparison of experimental phase portraits with the phase portraits from the approximate-small simulation (approximately linear with small lateral motions). Red lines correspond to simulated data; blue corresponds to experimental data. Data for experimental conditions matching Run 409.

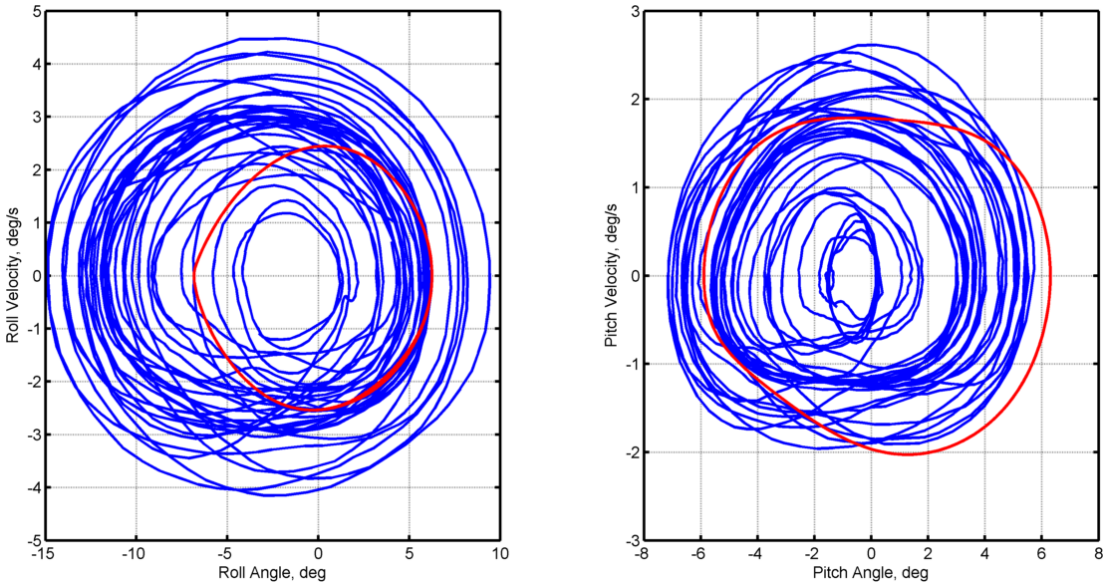


Figure A.7: Comparison of experimental phase portraits with the phase portraits from the approximate-large simulation (approximately linear with large lateral motions). Red lines correspond to simulated data; blue corresponds to experimental data. Data for experimental conditions matching Run 409.

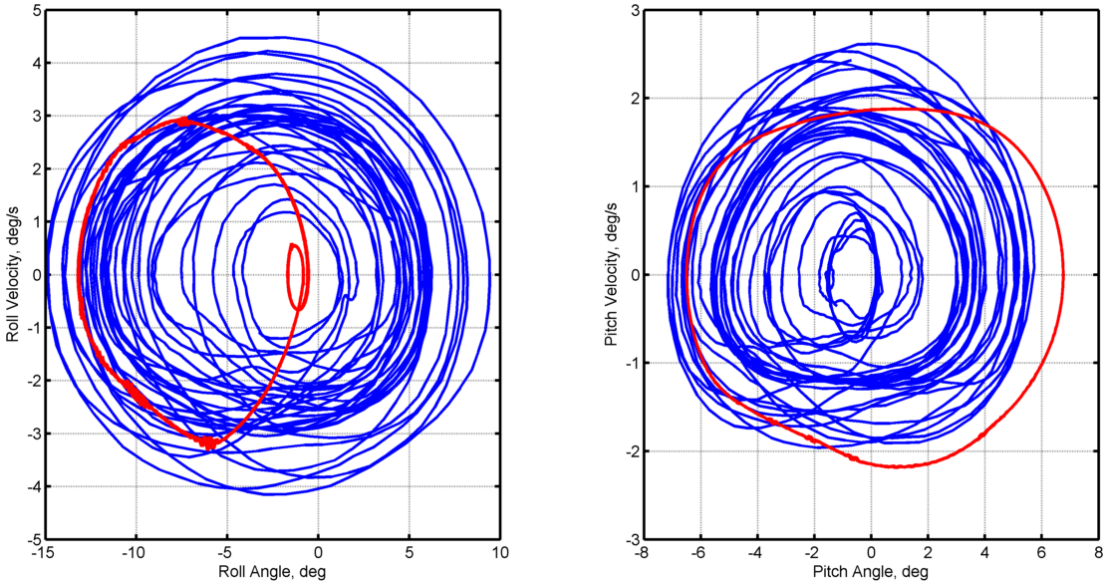


Figure A.8: Comparison of experimental phase portraits with the phase portraits from the fully nonlinear simulation. Red lines correspond to simulated data; blue corresponds to experimental data. Data for experimental conditions matching Run 409.

APPENDIX B

Additional ANOVA Comparisons of Validation Metric Performance

As mentioned in Chapter 3, ANOVA was used to perform comparisons of pairs of the calculation types, as well as the combined comparison of all three calculations types for which results are shown in the chapter. In other words, in addition to the results shown in Chapter 3, where ANOVA was run using data from all three code versions, ANOVA was also run only using data for each pair of code versions. This provides greater insight into the behavior of the metrics by focusing on the differences between specific code versions. The results from these additional ANOVA calculations are presented in this appendix.

First, results are shown for 2-D metric calculations. The results of an ANOVA analysis of only the linear and approximate-small code versions are shown in Figure B.1. Similar results for the linear and approximate-large code versions are shown in Figure B.2. Both of these analyses show no particularly noteable differences compared to the combined results comparing all three code versions (shown in Chapter 3). In contrast, an investigation of just the approximate-small and approximate-large metrics, shown in Figure B.3, showed a disagreement between the sample entropy and

the other metrics. While the other metrics (particularly the Hurst and Lyapunov exponents) show results similar to the other 2-D investigations, the sample entropy (and to a lesser extent, the correlation dimensions) values were very similar for the two code versions, with only about thirty percent of the wave conditions producing results which had statistically significant differences. The conclusions drawn from this data is that these metrics are not always equally effective and it is best to use a suite of metrics to validate chaotic time series simulations. Generally speaking, a conclusion that two attractor shapes are the same should be based upon investigation with multiple validation metrics; these results suggest that a single metric should not be considered sufficient.

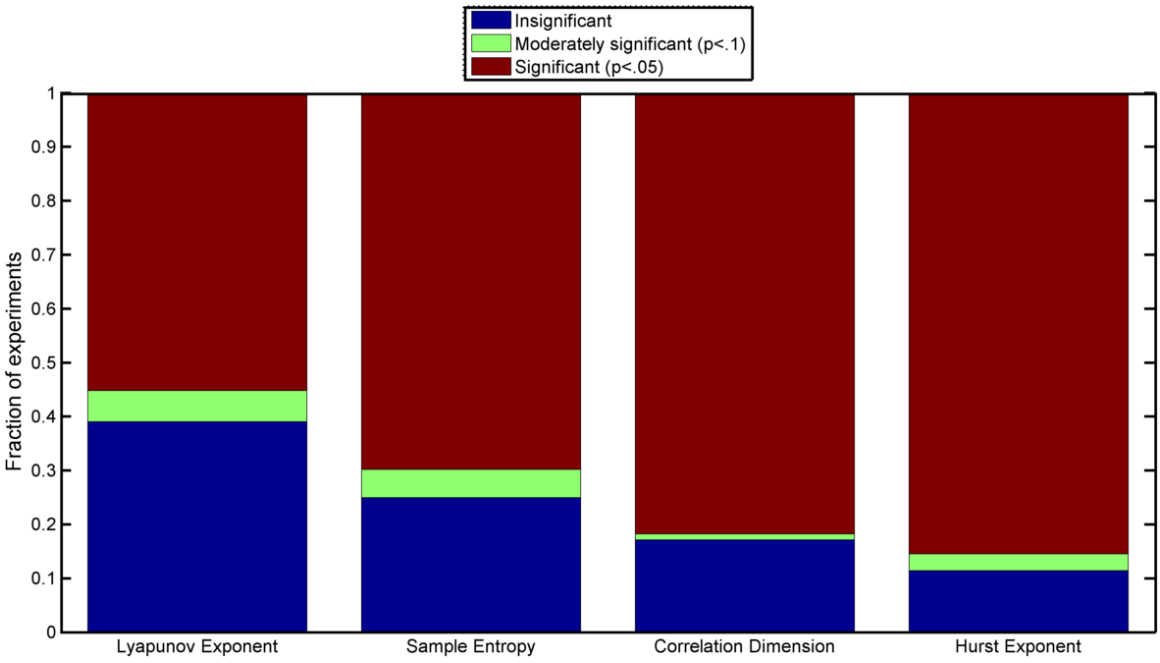


Figure B.1: Classification of the statistical significance associated with metric values calculated using 2-D data for each set of wave conditions. Results shown a comparison of the linear and approximate-small code versions.

The fraction of the conditions with statistically significant differences in the 4-D data is consistently high; no behavior such as that observed for the sample entropy in Fig-

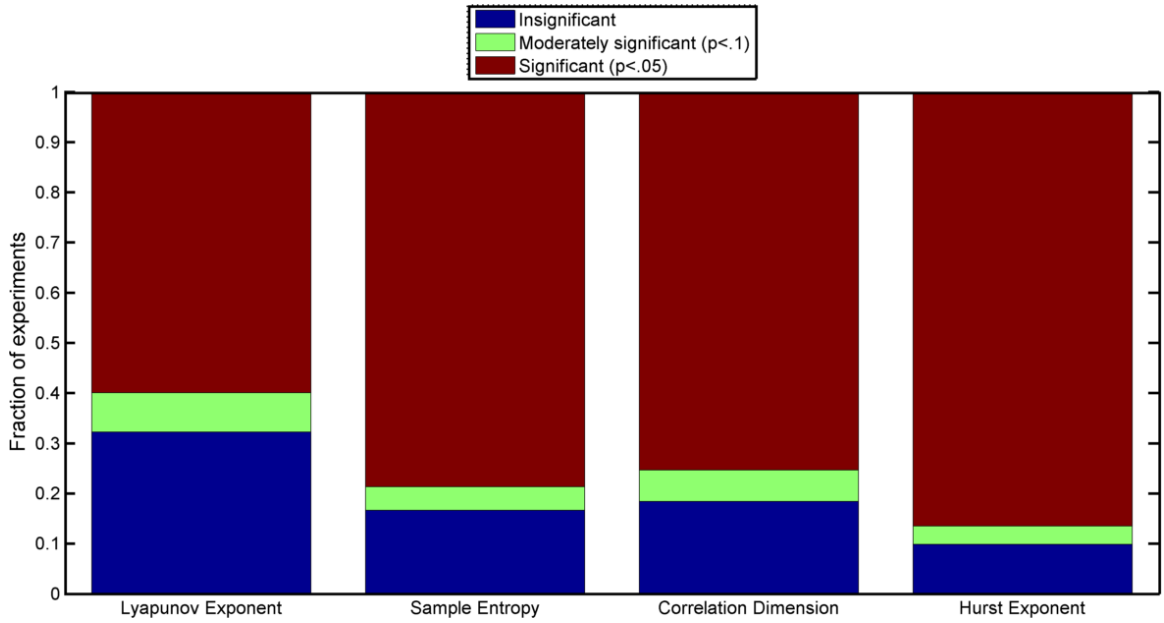


Figure B.2: Classification of the statistical significance associated with metric values calculated using 2-D data for each set of wave conditions. Results shown a comparison of the linear and approximate-large code versions.

ure B.3 is observed in the 4-D data. This may indicate that by virtue of containing more of the solution space, the 4-D metrics are more robust than the 2-D metrics. The results based upon the linear and approximate-small code versions are shown in Figure B.4, while the results comparing the linear and approximate-large code version are shown in Figure B.5, and the results comparing the approximate-small and approximate-large code versions are shown in Figure B.6.

Taken together, these results are similar to the combined results shown in Chapter 3. However, there is noteworthy variability between the metric results for different comparisons. This is most notable for the sample entropy and the correlation dimensions. The Hurst and Lyapunov exponents, in contrast, were more consistent over this data set. The key observation here, however, is that these metrics are not equally effective under all conditions. As described previously, sample entropy did not perform opti-

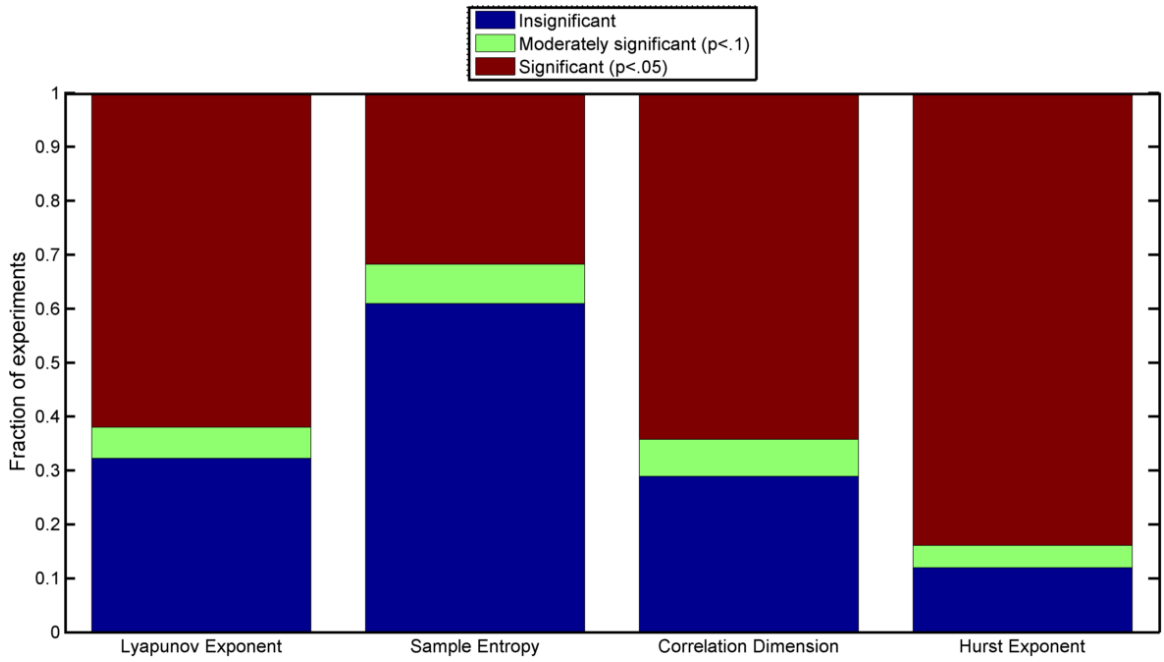


Figure B.3: Classification of the statistical significance associated with metric values calculated using 2-D data for each set of wave conditions. Results shown a comparison of the approximate-small and approximate-large code versions.

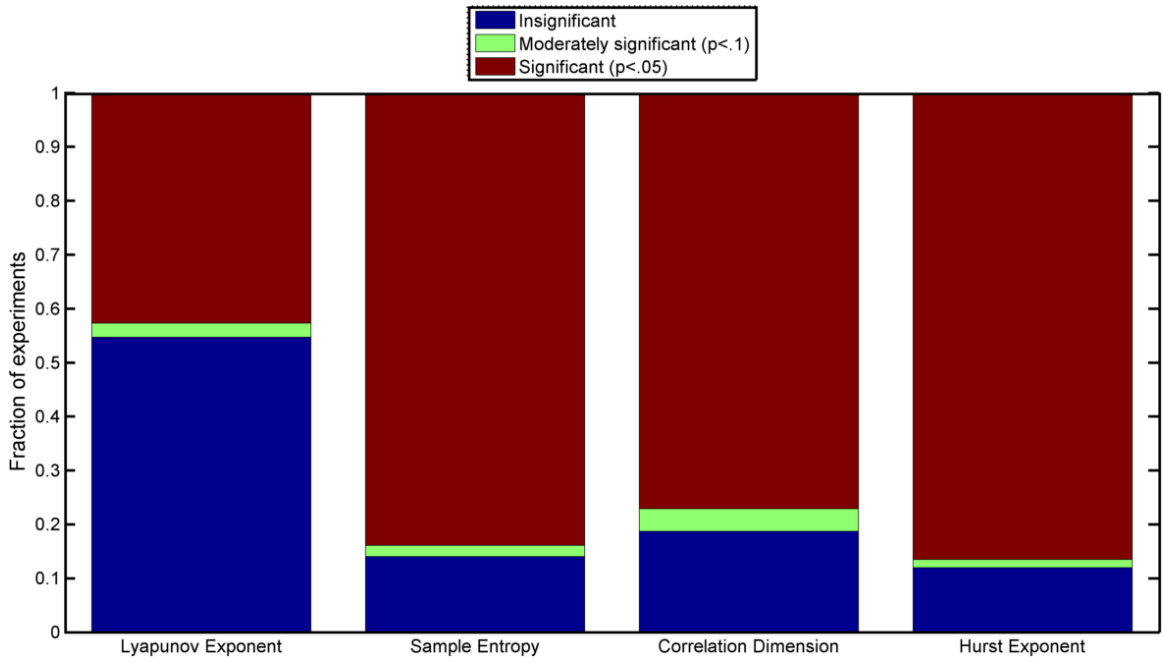


Figure B.4: Classification of the statistical significance associated with metric values calculated using 4-D data for each set of wave conditions. Results shown a comparison of the linear and approximate-small code versions.

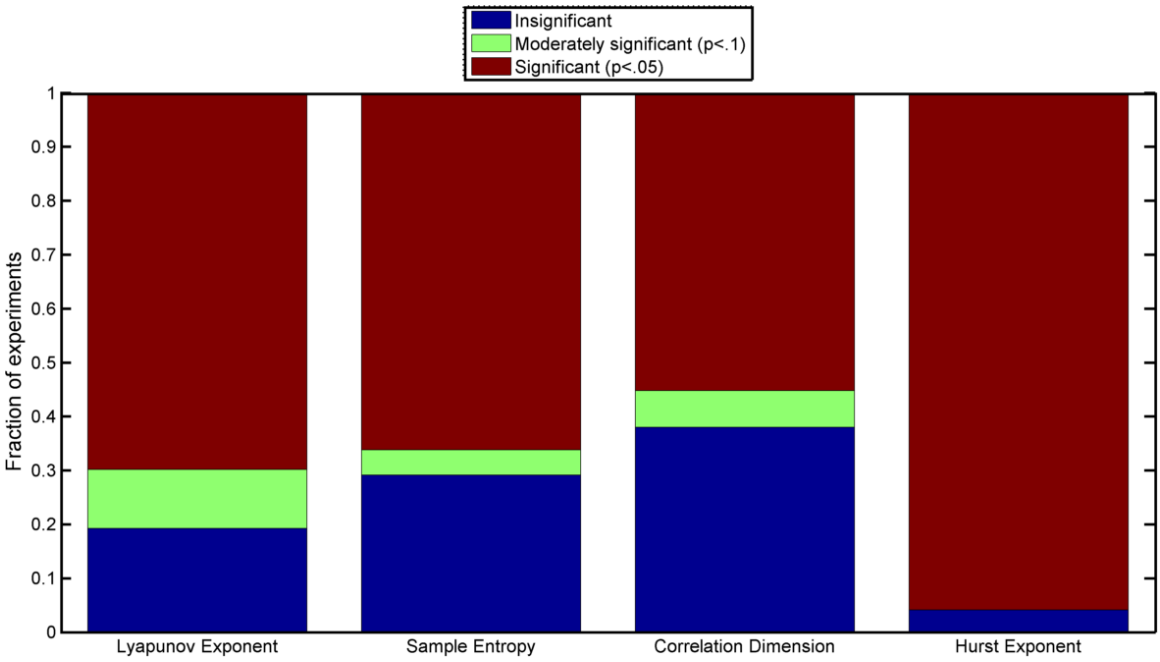


Figure B.5: Classification of the statistical significance associated with metric values calculated using 4-D data for each set of wave conditions. Results shown a comparison of the linear and approximate-large code versions.

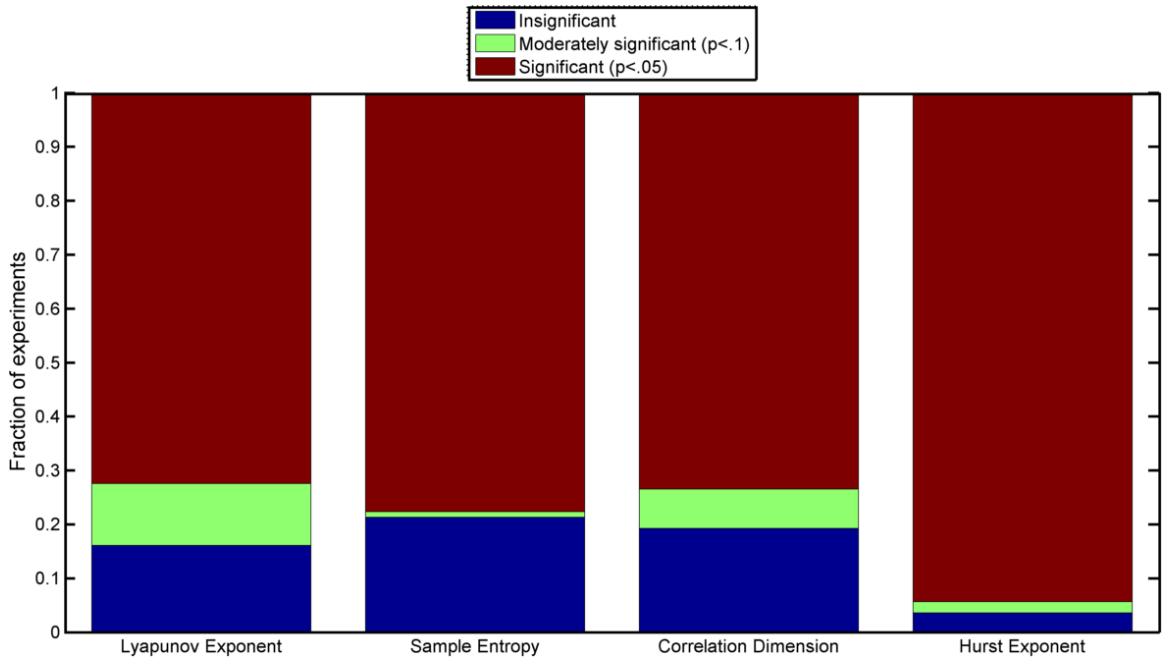


Figure B.6: Classification of the statistical significance associated with metric values calculated using 4-D data for each set of wave conditions. Results shown a comparison of the approximate-small and approximate-large code versions.

mally in the 2-D comparison of the approximate-small and approximate-large code versions. For other comparisons, it performed quite well. The conclusions drawn here is not that the sample entropy is a poor metric, but rather that validation of chaotic time series is best done using a suite of validation metrics. Substantially more information, and correspondingly confidence, can be obtained by using multiple validation metrics, rather than relying on a single metric.

APPENDIX C

POD ANOVA Results

The results for the POD metric are presented and discussed in this appendix. For these results, trajectories were referenced to the mean fundamental mode of the linear results for each wave condition, unlike Section 3.3.2 where the values were calculated based on a nonlinear reference. For a full explanation on how metrics are calculated refer back to Section 2.2.2 in Chapter 2 for a detailed explanation of how POD metric values are calculated. Additionally, note that only 4-D results are presented here. The reason for this is that higher order modes appear to be more sensitive than lower order modes. For the 2-D case, there is only one independent mode (the second mode is fully defined by the first mode). For the 4-D case, on the other hand, there are three meaningful modes, with only the fourth and final mode being fully defined by the other modes. Figure C.1 shows the results including all three code versions. Results for the combinations of linear and approximate-small, linear and approximate-large, and approximate-small and approximate-large are shown in Figures C.2, C.3, and C.4, respectively.

There are several observations to be drawn from the POD results. It is apparent that all results including the approximate-large code version show strong, statistically

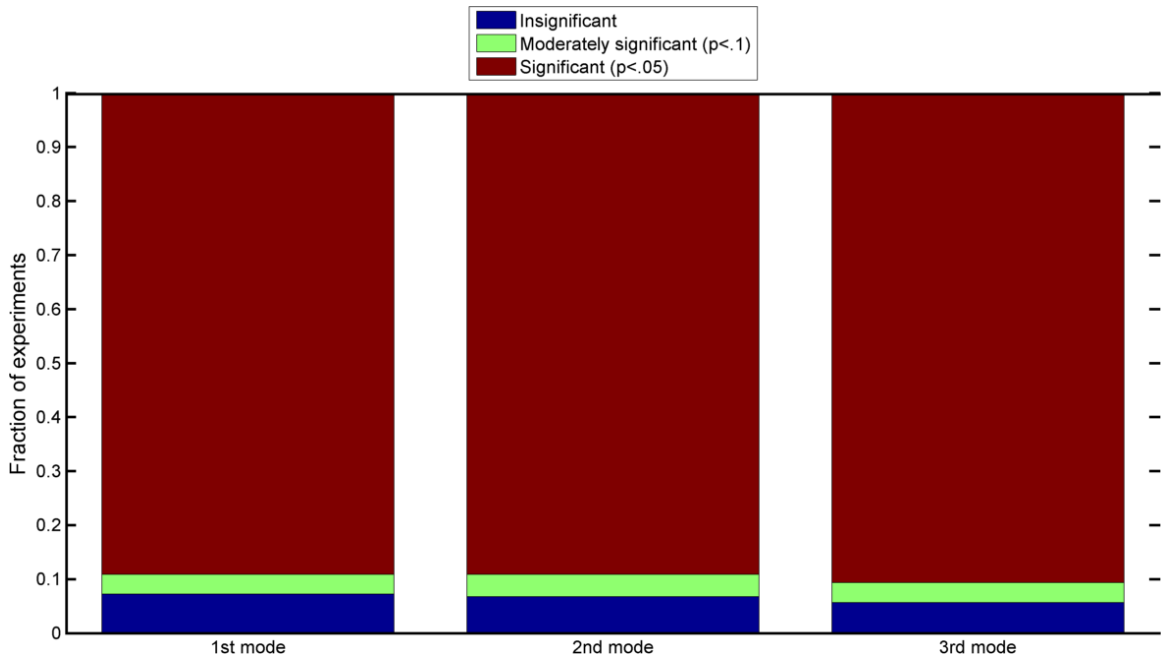


Figure C.1: Classification of the statistical significance associated with POD metric values calculated using 4-D data for each set of wave conditions. Results shown a comparison of all three code versions (linear, approximate-small, and approximate-large).

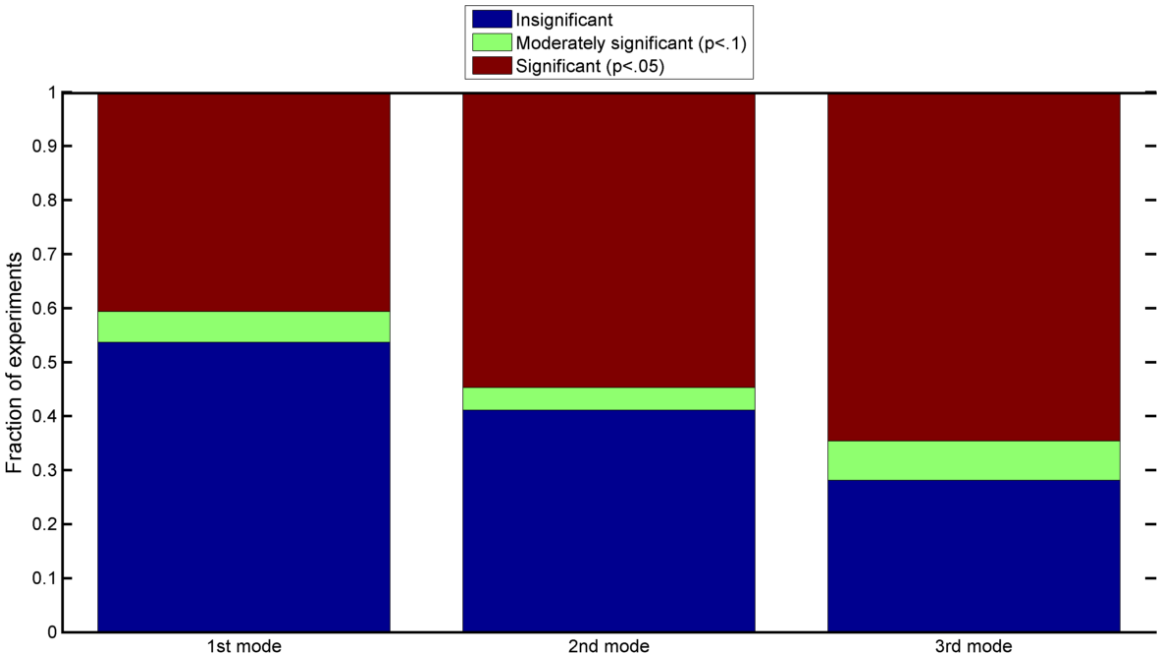


Figure C.2: Classification of the statistical significance associated with POD metric values calculated using 4-D data for each set of wave conditions. Results shown a comparison of the linear and approximate-small code versions.

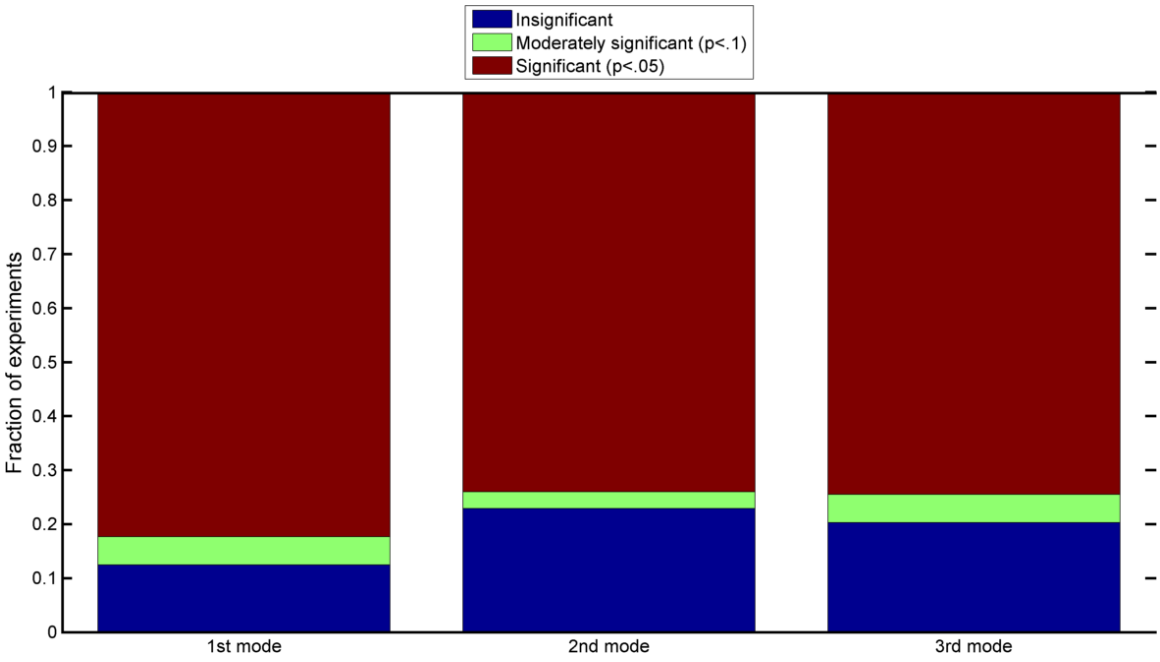


Figure C.3: Classification of the statistical significance associated with POD metric values calculated using 4-D data for each set of wave conditions. Results shown a comparison of the linear and approximate-large code versions.

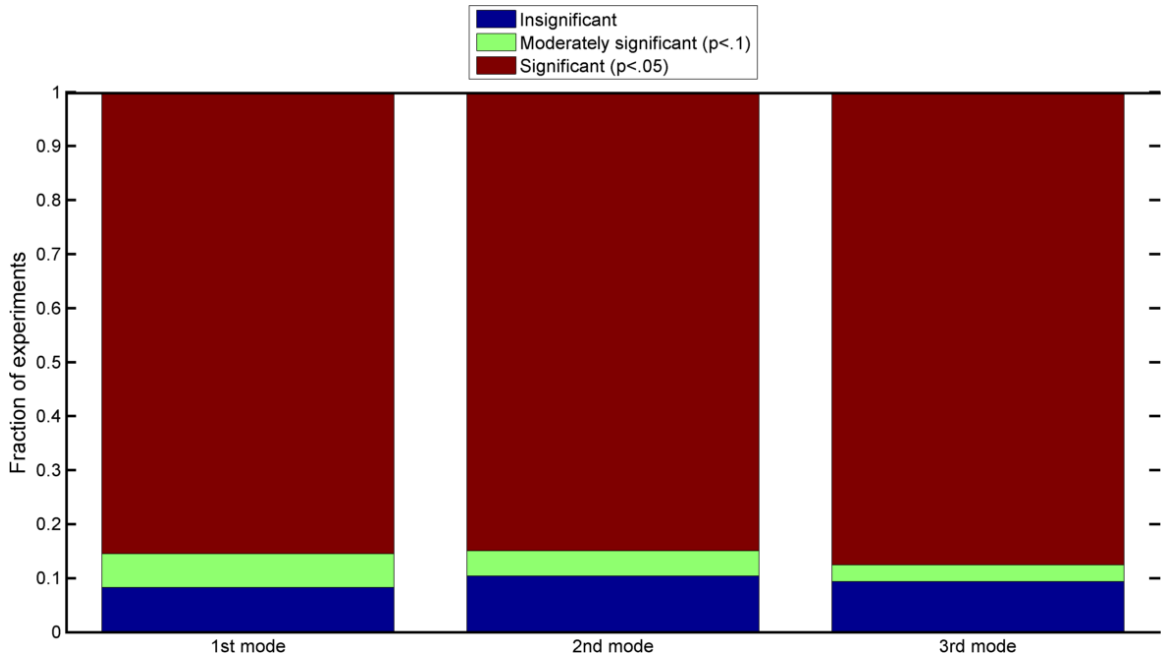


Figure C.4: Classification of the statistical significance associated with POD metric values calculated using 4-D data for each set of wave conditions. Results shown a comparison of the approximate-small and approximate-large code versions.

significant differences. However, Figure C.2 shows much lower levels of statistical significance by comparison. From examination of the data, and the results for the other metrics, it is known that the approximate-large results have a tendency towards capsize, making the behavior of this code version significantly different from the linear and approximate-small code versions, which have comparatively subtle differences. When comparing these two code versions, the POD metric (applied to the fundamental POD mode) shows a relative insensitivity to the physics when compared to the other candidate metrics, as comparatively few statistically significant differences are found between these metrics. It is interesting to note, however, that the performance of the POD metric improves noteably as higher order modes are considered. Although more research should be done, this suggests that POD provides a hierarchy of sensitivity to differences in physics. The fundamental mode can be used to investigate gross differences in physics, while higher order modes may be used to detect increasingly subtle differences in physics. This suggests an intriguing avenue for future research.

BIBLIOGRAPHY

Abramowitz, M. and Stegun, I. (1972). "Handbook of Mathematical Functions: with Formulas, Graphs, and Mathematical Tables (10th Printing with corrections)." Dover Publications, New York, USA.

AIAA (1998). Guide for the Verification and Validation of Computational Fluid Dynamics Simulations." American Institute of Aeronautics and Astronautics, Reston, VA. G-077-1998.

Arnold, L., Chueshov, I., and Ochs, G. (2003). "Stability and capsizing of ships in random sea- a survey." Tech. rept. 464. Universitat Bremen Insitut fur Dynamicshe Systeme.

ASME Performance Test Code Committee 60. (2006). Guide for verification and validation in computational solid mechanics." American Society of Mechanical Engineers, New York, NY. ASME V&V 10-2006.

Atanasoff, J.V. (1940). "Computing Machine for the Solution of large Systems of Linear Algebraic Equations." In: "The Origins of Digital Computers" Ed. Randell, B. (1982). Springer, New York.

Battistin, D. and A. Iafrati (2003). "Hydrodynamic loads during water entry of two-dimensional and axisymmetric bodies." Journal of Fluids and Structures, Vol. 17, Is. 5, pp. 643-664.

Battistin, D. and A. Iafrati (2004). "A numerical model for the jet flow generated by water impact." Journal of Engineering Mathematics, Vol. 48, Is. 3, pp. 353-374.

Blair, D. and E. Durfesne. "The Matlab Particle Tracking Code Repository." from physics.georgetown.edu/matlab/

Bibuli, M., Bruzzone, G., Caccia, M. and Lapierre, L. (2009). "Path-following algorithms and experiments for an unmanned surface vehicle." Journal of Field Robotics. Vol. 26.

Breivik, M., Hovstein, V. and Fossen, T. (2008). "Straight-Line Target Tracking for Unmanned Surface Vehicles." Modeling, Identification, and Control, Vol. 29.

Bulian, G. (2005). "Nonlinear parametric rolling in regular wavesa general procedure for the analytical approximation of the GZ curve and its use in time domain simulations." Ocean Engineering, Vol. 32, Is. 34, pp. 309-330.

Caccia, M., Bibuli, M., Bono, R. and Bruzzone, G. (2008). "Basic Navigation, Guidance and Control of an Unmanned surface Vehicle." Autonomous Robots, Vol. 25.

Cameron, R. and Martin, W. (1947). "The orthogonal development of nonlinear functionals in series of Fourier-Hermite functionals." Ann Math, Vol. 48, Is. 2, pp. 385-392.

Carcaterra, A. and E. Ciappi (2004). "Hydrodynamic shock of elastic structures impacting on the water: theory and experiments." Journal of Sound and Vibration 271(1-2): 411-439.

Chen, C., and Jaw, S. (1998). "Fundamentals of turbulence modeling." Taylor & Francis, Washington, DC.

Chui, C. and Chen, G. (1989). "Linear systems and optimal control." Springer Berlin Heidelberg.

Colwell, J.L. (2002). Real time ship motion criteria for maritime helicopter operations. ICAS 2002 Congress.

Cooker, M.J. and Peregrine, D.H. (1995). "Pressure-impulse theory for liquid impact problems." Journal of Fluid Mechanics, Vol. 297, Is. 1, pp. 193-214.

Cooper, M. and McCue, L. (2011). Experimental study on deformation of flexible wedge upon water entry. 9th Symposium on High Speed Marine Vehicles, Naples, Italy.

Cooper, M.and McCue, L. (2012). Effectiveness of chaotic system measures for the validation of ship dynamics simulations. 11th International Conference on the Stability of Ships and Ocean Vehicles, Athens, Greece.

Cooper, M., Wu, W., and McCue, L.S. (2015). "Non-intrusive polynomial chaos for efficient uncertainty analysis in parametric roll simulations." Submitted to the Journal of Marine Science and Technology for review.

Cooper, M., and McCue, L.S. (2015). "Design of a controller for autonomous vessel recovery utilizing the prediction of host vessel motion." Accepted with minor revisions by Naval Engineers Journal.

De Backer, G.,Vantorre, M., et al. (2009). "Experimental investigation of water impact on axisymmetric bodies." Applied Ocean Research, Vol. 31, Is. 3, pp. 143-156.

Department of Defense (2009). DoD Modeling and SimulatioInn (M & S) Verification, Validation, and Accreditation (VV&A). Department of Defense Instruction 5000.61.

Department of the Navy (1999). Verification, Validation, and Accreditation (VV&A) of Models and Simulations. SECNAV Instruction 5200.40.

Faltinsen, O.M. (2000). "Hydroelastic slamming." Journal of Marine Science and Technology, Vol. 5, Is. 2, pp. 49-65.

Falzarano J.M., Shaw, S.W., and Troesch, A.W. (1992). "Application of global methods for analyzing dynamical systems to ship rolling motion and capsizing." International Journal of Bifurcation and Chaos, Vol. 2, Is. 1, pp. 101-115.

Federico, F. and Amoruso A. (2009). "Impact between fluids and solids. Comparison between analytical and FEA results." International Journal of Impact Engineering, Vol. 36, Is. 1, pp. 154-164.

Ferrier, B., Applebee T., Manning, A., and James, D. (2000). Landing period designator visual helicopter recovery aide; theory and real-time application. Proceedings of the 56th Annual Forum of the American Helicopter Society.

Ferrier, B., Baitis A.E., and Manning, A. (2000). Evolution of the landing period designator (LPD) for shipboard air operations. Naval Engineers Journal, Vol. 112, Is. 4, pp. 297-315.

Ferrier, B. and Manning, A. (1998). Simulation and testing of the landing period designator (LPD) helicopter recovery aid. Naval Engineers Journal, Vol. 110, Is. 1, pp. 189-205.

France, W., Levadou, M., Treakle, T., Paulling, J., Michel, R., and Moore, C. (2001). "An Investigation of Head-Sea Parametric Rolling and its Influence on Container Lashing Systems." SNAME Annual Meeting 2001.

France, W.N., Levadou, M., Treakle, T.W., Paulling, J.R., Michel, R.K., and Moore, C. (2003). "An Investigation of Head-Sea Parametric Rolling and its Influence on Container Lashing Systems." Marine Technology, Vol. 40, Is. 1, pp. 1-19.

Gaudet, S., A. Iafrati, et al. (1998). "Numerical simulation of circular disks entering the free surface of a fluid. Initial stage of flat plate impact onto liquid free surface." Physics of Fluids, Vol. 10, Is. 10, pp. 2489-2499.

Ghanem, R. and Spanos P. (1991). "Stochastic Finite Elements: a Spectral Approach." Springer-Verlag New York, Inc., New York, USA.

Ghanem, R. (1999). "Ingredients for a general purpose stochastic finite element formulation." Comp Meth Appl Mech Eng, Vol. 168, pp. 19-34.

Gougoulidis, G. (2008). "The Utilization of Artificial Neural Networks in Marine

Applications: An Overview." Naval Engineers Journal. Vol. 120, No. 3, pp. 19-26.

Grassberger P. and I. Procaccia. (1983). Characterization of Strange Attractors. Physical Review Letters, Vol. 50, Is. 5, pp. 346-349.

Haddara, M. and Xu, J. (1999). "On the identification of ship coupled heave-pitch motions using neural networks." Ocean Engineering. Vol. 26, pp. 381-400.

Hayden, D., Bishop, R., Park, J. and Laverty, S. (2006). "Model 5514 Capsize Experiments Representing the Pre-Contract DDG51 Hull Form at End of Service Life Conditions." NSWCCD-50-TR2006/020, Hydromechanics Department Report. Carderock Division, Naval Surface Warfare Center.

Hess, D., Faller, W., Fu, T. and Ammeen, E. (2006). "Improved Simulation of Ship Maneuvers Using Recursive Neural Networks." The 44th AIAA Aerospace Sciences Meeting, Reno, NV, pp. 2006-1481.

Hess, D., Faller, W., Fu, T. and Ammeen, E. (2006). "Ship Maneuvering Simulation in Wind and Waves: A Nonlinear Time-Domain Approach Using Recursive Neural Networks." The 26th Symposium on Naval Hydrodynamics, Rome, Italy.

Hosder, S., Walters, R.W., and Balch, M. (2010). "Point-collocation nonintrusive polynomial chaos method for stochastic computational fluid dynamics." AIAA, Vol. 48, Is. 12, pp. 2721-2730.

Hosder, S., Walters, R.W., and Perez, R. (2006). "A non-intrusive polynomial chaos method for uncertainty propagation in CFD simulations." 44th AIAA Aerospace Sciences Meeting and Exhibit, Reno, Nevada.

Ioannou, P. and Sun, J. (2012). "Robust Adaptive Control." Dover Publications. Khan, A., Bil, C. and Marion, K. (2005). Ship Motion Prediction for Launch and Recovery of Air Vehicles. OCEANS, 2005.

Judge C.T.A. and Perlin M. (2004). "Initial water impact of a wedge at vertical and oblique angles." J. Eng. Math., Vol. 48, Is. 3/4.

Kim, Y. editor (2010). Proceedings of the ITTC Workshop on Seakeeping -V&V for Non-linear Seakeeping Analysis, Seoul, Korea, October.

Korobkin, A. (2004). "Analytical models of water impact." European Journal of Applied Mathematics, Vol. 15, Is. 6, pp. 821-838.

Korobkin, A. (2007). "Second-order Wagner theory of wave impact." Journal of Engineering Mathematics, Vol. 58, Is. 1, pp. 121-139.

Korobkin, A. A. and A. Iafrati (2005). "Hydrodynamic loads during initial stage of floating body impact." Journal of Fluids and Structures, Vol. 21, Is. 4, pp. 413-427.

Lainiotis, D., Plataniotis, K., Menon, D. and Charalampous, C. (1993). "Adaptive Heave Compensation Via Dynamic Neural Networks." Oceans. Vol. 1, pp. 243-248.

Lake, D.E., Richman, J.S., Griffin, M.P., and Moorman, J.P. (2002). "Sample entropy analysis of neonatal heart rate variability." American Journal of Physiology - Regulatory, Integrative and Comparative Physiology, Vol. 283, Is. 3, pp. R789-R797.

Leonessa, A., VanZwieten, T. and Morel, Y. (2006). "Neural Network Model Reference Adaptive Control of Marine Vehicles." Current Trends in Nonlinear Systems and Control Systems and Control: Foundations & Applications.

Lucor, D., Xiu, D. and Karniadakis, G.E. (2001). "Spectral representations of uncertainty in simulations: Algorithms and applications." In: Proceedings of the International Conference on Spectral and High Order Methods, Uppsala, Sweden.

Luit, D.A., Mook, D., VanLandingham, H.F., and Nayfeh, A.H. (2000). "Roll reduction in ships by means of active fins controlled by a neural network." Ship Technology Research, pp. 79-89.

Manley, J. (2008) "Unmanned Surface Vehicles, 15 Years of Development." OCEANS 2008.

MATLAB version 2013a Neural Network Toolbox. (2013). The MathWorks Inc. Natick, Massachusetts.

McCue, L.S. (2008). Motion prediction envelopes for intact and damaged hulls. Proceedings of the 10th International Ship Stability Workshop, Daejeon, Korea.

McCue, L.S., Bassler, C., and Belknap, W. (2006). "Real-Time Identification of Behavior Leading to Capsize." Procedings of the 9th International Conference on the Stability of Ships and Ocean Vehicles, Rio de Janeiro, Brazil.

McCue, L.S., Belknap, W., and Campbell, B. (2006). Reconciling experimental and numerical data: techniques of nonlinear seakeeping code validation. invited submission to Marine Systems and Ocean Technology, Vol. 2, Is. 1-2, pp. 55-62. Originally presented at the 8th International Ship Stability Workshop, Istanbul, Turkey, October, 2005.

McCue, L., W. R. Story, and A. Reed (2008). Nonlinear dynamics applied to the validation of computational methods. Proceedings of 27th Symposium on Naval Hydrodynamics, Seoul, Korea, October.

McCue, L.S. and Troesch, A.W. (2004). "Use of Lyapunov Exponents to Predict Chaotic Vessel Motions." Proceedings of the 7th International Ship Stability Workshop, Shanghai, China.

McCue, L.S., Wu, W., Story, W.R., and Xing, Z. (2009). Modeling, verification, and validation of chaotic systems with application to ship motions and the Lorenz oscillator. Proceedings of 2009 NSF Engineering Research and Innovation Conference, Honolulu, HI, June.

Metropolis, N. (1987). "The beginning of the Monte Carlo method." Los Alamos Science (1987 Special Issue dedicated to Stanislaw Ulam), pp. 125130.

Nomoto, K., Taguchi, T., Honda, K. and Hirano, S. (1957). "On the steering qualities of ships." Technical report, International Shipbuilding Progress, Vol. 4.

Oberkampf, W.L. and Roy, C.J. (2010). "Verification and Validation in Scientific Computing." New York, Cambridge University Press.

Oger, G., M. Doring, et al. (2006). "Two-dimensional SPH simulations of wedge water entries." Journal of Computational Physics, vol. 213, Is. 2, pp. 803-822.

Okada, S. and Y. Sumo (2000). "On the water impact and elastic response of a flat plate at small impact angles." Journal of Marine Science & Technology, Vol. 5, Is. 1, pp. 31.

O'Reilly, Peter J.F. (1987). Aircraft/deck interface dynamics for destroyers. Marine Technology, Vol. 24, Is. 1, pp. 15-25.

Ott, L. and Longnecker, M. (2001) "An Introduction to Statistical Methods and Data Analysis." 5th edition. Duxbury, Pacific Grove, CA.

Panciroli R. (2013). "Water entry of flexible wedges: Some issues on the FSI phenomena." Applied Ocean Research, Vol. 39, pp. 24-74.

Panciroli R., Abrate, S., Minak, G. (2013). "Dynamic response of flexible wedges entering the water." Composite Structures, Vol. 99, pp. 163-171.

Papoulias, F.A. (1987). "Dynamical analysis of mooring systems." Doctoral Dissertation. University of Michigan, Ann Arbor.

Paulling, J.R. (2006). "Parametric Rolling of Ships Then and Now." Proceedings of the 9th International Conference on Stability of Ships and Ocean Vehicles', Rio de Janeiro, Brazil.

Pearson, K. (1901). On Lines and Planes of Closest Fit to Systems of Points in Space.

Philsophical Magazine Vol. 2, Is. 6, pp. 559-572.

Rebollo, T. and Lewandowski, R. (2014). "Mathematical and Numerical Foundations of Turbulence Models and Applications." Springer, New York.

Reed, A. (2009). A Naval Perspective on Ship Stability. Proceedings of the 10th International Conference on Stability of Ships and Ocean Vehicles, Russia.

Roache, P.J. (1998). "Verification and Validation in Computational Science and Engineering." Hermosa, Albuquerque, NM.

Roberts, G. and Sutton, R. Eds. (2006). "Advances in Unmanned Marine Vehicles." Institution of Electrical Engineers, UK.

Rugh, W. (1993). "Linear System Theory." Prentice Hall.

Sano, M. and Sawada Y. (1985). "Measurement of Lyapunov Spectrum from a Chaotic Time Series." Physical Review Letters, Vol. 55, Is. 10.

Soliman, M.S. and Thompson J.M.T. (1991). "Transient and steady state analysis of capsize phenomena." Applied Ocean Research, Vol. 13, Is. 2, pp. 82-92.

Sonnenburg, C. (2012). "Modeling, Identification, and Control of an Unmanned Surface Vehicle." Doctoral Dissertation. Virginia Polytechnic Institute and State University.

Spyrou, K.J. (1996). "Homoclinic connections and period doublings ofa ship advancing in quartering waves." Chaos, Vol. 6, Is. 2, pp. 209-218.

Story, W.R. (2009). "Application of Lyapunov Exponents to Strange Attractors and Intact & Damaged Ship Stability." Masters Thesis. Virginia Polytechnic Institute and State University.

Story, W.R., Xing, Z., Wu, W., and McCue, L.S. (2010). Validation: a historical look and two suggested techniques. Proceedings of ITTC Workshop on Seakeeping V&V for Non-linear Seakeeping Analysis, Seoul, Korea, October.

Ulstein, T. and O. Faltinsen (1994). "Hydroelasticity analysis of a flexible bag structure." Proceedings of the 20th Symposium on Naval Hydrodynamics, University of California, Santa Barabara, National Academy Press, Washington, D.C.

Ulstein, T. and O. Faltinsen (1998). "Cobblestone effect on SES. Symposium on Fluid Dynamics Problems of Vehicles Operating near or in the Air-Sea Interface." Amsterdam, Research and Technology Agency, NATO, Neuilly-Sur-Seine Cedex, France.

US Navy (2007). "The Navy Unmanned Surface Vehicle (USV) Master Plan."

Weiner, N. (1938). "The homogeneous chaos." Am J Math, Vol. 60 Is. 4, pp. 897-936.

Wolfram S. (2003). "The Past and Future of Scientific Computing." Presentation at the International Symposium on Modern Computing.

Wu, W., Bulian, G. and McCue, L.S. (2011). "Uncertainty Analysis for Parametric Roll Using Non-intrusive Polynomial Chaos." Proceedings of the 12th International Ship Stability Workshop, Washington DC, USA.

Wu, W., McCue, L.S., and Roy, C. (2011). "The Method of Manufactured Solutions Applied to Chaotic Systems." Nonlinear Dynamics, Vol. 63, Is. 4, pp. 667-679.

Xing, Z. and McCue, L. (2009). "Parameter identification for two nonlinear models of ship rolling using neural networks." Proceedings of the 10th International Conference on Stability of Ships and Ocean Vehicles. St. Petersburg, Russia, 421-428.

Xing, Z. and McCue, L. (2010). "Modeling ship equations of roll motion using neural networks." Naval Engineering Journal. Vol. 3, pp. 49-60.

Xiu, D. and Karniadakis, G.E. (2002). "The Wiener-Askey polynomial chaos for stochastic differential equations." SIAM J Sci Comput, Vol. 24, Is. 42, pp. 619-644.

Xiu, D. and Karniadakis, G. (2003). "Modeling uncertainty in flow simulations via generalized polynomial chaos." J Comput Phys, Vol. 187, Is. 1, pp. 137-167.

Yee, H.C., Sweby, P.K. (1994). Global asymptotic behavior of iterative implicit schemes. Int. J. Bifurc. Chaos, Vol. 4, Is. 6, pp. 1579-1611.

Yee, H.C., Sweby, P.K. (1996). Nonlinear dynamics and numerical uncertainties in CFD. Technical Report. NASA Technical Memorandum 110398, NASA, April.

Yee, H.C. and Sweby, P.K. (1997). Dynamics of numerics and spurious behaviors in CFD computations. Technical Report RIACS 97.06, Research Institute for Advanced Computer Science NASA Ames Research Center, June.

Yee, H.C., Torczynski, J.R., Morton, S.A., Visbal, M.R., and Sweby, P.K. (1997). On spurious behavior of CFD simulations. Proceedings of 13th AIAA Computational Fluid Dynamics Conference, Snowmass, CO.

Yettou, E.-M., A. Desrochers, et al. (2006). "Experimental study on the water impact of a symmetrical wedge." Fluid Dynamics Research, Vol. 38, Is. 1, pp. 47-66.