Data-Driven Propeller Modeling for Ship Maneuvering

by

Bradford G. Knight

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Naval Architecture and Marine Engineering)
in The University of Michigan
2021

Doctoral Committee:

Associate Professor Kevin J. Maki, Chair
Associate Professor Matthew D. Collette
Professor Joaquim R. R. A. Martins
Professor Yin Lu Young

Bradford G. Knight

bgknight@umich.edu

ORCID iD: 0000-0003-0038-6487

I dedicate this work to my family.

ACKNOWLEDGEMENTS

I would like to thank the many people who have supported me throughout the PhD process including but not limited to the many professors at the University of Michigan with whom I have taken classes, my peers in my classes and research group, and my PhD committee. I would especially like to thank my PhD advisor Kevin Maki for his mentorship and guidance. Additionally, I would like to thank my family for their love and support.

I am also very appreciative of the Office of Naval Research as well as the Ford-UofM alliance whom have provided funding for my PhD.

TABLE OF CONTENTS

LIST OF FIGURES

Figure

LIST OF TABLES

Table

LIST OF APPENDICES

Appendix

ABSTRACT

The analysis and understanding of the maneuvering characteristics of seafaring vessels is critical for their successful operation. Ships operate in the harsh ocean environment and must be able to either withstand the storms that they sail through or alter course to avoid particularly severe storms. As vessels become more automated and design practices shift toward reducing emissions it is important to gain an understanding of the expected maneuvering characteristics of vessels both in calm water and in waves. One way to determine the maneuvering characteristics of a vessel is by using Computational Fluid Dynamics (CFD) to account for effects like transient flow separation over appendages and green water over the deck of the ship. Both the propeller and the rudder are critical components of this analysis, but it is difficult to accurately and efficiently determine the forces generated by these surfaces. The differences in time and length scales between the propeller and the hull create a bottleneck for efficiently modeling a vessel with numerical methods since the limiting time step size is that of the propeller. There is complex flow interaction between the hull, propeller, and rudder which is important to capture to properly model the multi-dimensional forces on each surface.

One way to alleviate the cost of directly modeling the rotating propeller is to apply a propeller model. Existing propeller models often use simplifying assumptions and may predict incorrect forces especially in off-design conditions. A propeller model is useful to reduce computational cost, but it is desirable for the propeller model to be as accurate as a high-fidelity method.

The objective of this work is to develop a framework for a data-driven propeller

and rudder model that can predict the forces with sufficient accuracy, such that the propeller model maintains the accuracy of a high-fidelity method but can be implemented in a maneuvering simulation at a significantly reduced cost. The data-driven model is trained with CFD simulations of the propeller operating in the behind condition with the rudder deflected to account for the flow interaction between the hull, propeller, and rudder. Different data-driven techniques are compared and evaluated. The rudder forces are also directly modeled by the data-driven model.

The data-driven propeller and rudder model is trained, evaluated, and implemented for two different model scale vessels. The first demonstration is on the KRISO Container ship which is a single screw model container ship with a semi-horned rudder. The second case study is on the twin-screw twin-rudder Office of Naval Research Tumblehome model scale surface combatant. The accuracy of the data-driven model is evaluated with different techniques and the accuracy is considered in terms of the underlying discretization and turbulence modeling uncertainty. Maneuvering simulations with the data-driven propeller and rudder model are performed for each vessel performing turning circle maneuvers in calm water and in waves.

This work demonstrates how a data-driven propeller and rudder model can be trained, validated, and implemented in a CFD maneuvering simulation. The computational cost is significantly reduced in both case studies performed and promising results are found.

CHAPTER I

Introduction

Seafaring vessels are critical for the transportation of goods and people to support the global economy. The marine industry must adapt the design of vessels to reduce greenhouse gas emissions to help combat climate change in the coming years. Vessels are also becoming more autonomous. These societal drivers will lead to changes in how vessels are designed and operated.

Vessels operate in harsh seaways and in storms. The maneuvering capabilities of a vessel must be understood to properly operate the vessel to prevent damage and to ensure the safety of crew and/or passengers on board. An alternative is to reroute ships to avoid severe storms which lengthens the voyage and increases emissions as well as costs. If the maneuvering characteristics of a vessel are better understood it not only leads to a safer vessel but also more efficient operational guidelines.

To assess the maneuvering characteristics of a vessel in waves it is important to generate engineering methods that are accurate but are also inexpensive enough that multiple simulations can be performed to gain a statistical understanding of how a vessel behaves in various sea conditions. Data-driven models that are trained with high fidelity simulations can enable fast calculations that are also accurate.

One aspect that makes modeling the maneuvering vessel with numerical methods particularly challenging is accurately determining the forces on both the propeller and

the rudder at a reasonable cost. The propeller and rudder operate in the wake of the vessel and the rudder also operates in the wake of the propeller. This complicated flow field is challenging to accurately resolve. Viscous numerical methods like Reynolds Averaged Navier Stokes Computational Fluid Dynamics (RANS CFD) can accurately determine the forces on the hull, propeller, and rudder (*Carrica et al.* (2007), *Cura-Hochbaum* (2006), *Shen et al.* (2015), *Tezdogan et al.* (2015)). These simulations model the turbulent two-phase flow but at a relatively high computational cost. The propeller is much smaller than the hull and rotates rapidly which leads to a significant difference in time and length scales. To model the rotating propeller with the hull in the same simulation requires that the vessel be modeled with the time step size of the propeller which significantly increases the cost. Even RANS CFD can be inaccurate for heavily separated flows, so numerical alternatives like Detached Eddy Simulation (DES) or Large Eddy Simulation (LES) are more expensive but are more suitable for heavily separated flows (*Breuer et al.* (2003); *Persson et al.* (2006)). An example of this phenomenon is when a propeller undergoes crashback (*Kumar and Mahesh* (2016); *Liao et al.* (2020); *Verma et al.* (2012)).

When a vessel maneuvers it has forward speed, but it also has sway velocity, yaw rate, and the propeller interacts with both the hull and rudder. In a maneuver, the vessel has six degrees of freedom and to correctly model the maneuvering trajectory the six degree of freedom force must be determined for the hull, propeller, and the rudder. Correctly calculating the propeller force is critical since the propeller thrust not only determines how much force drives the vessel, but also determines the inflow to the rudder which produces most of the turning moment.

The objective of this work is to maintain the accuracy of a high-fidelity RANS CFD simulation of a complete vessel with a discretized rotating propeller and rudder, but at a significantly reduced cost. The cost reduction is a factor of both the reduction in computational cost for the maneuvering simulation using the propeller model as well

as the means by which the model is trained to mitigate training expense. This work explores the use of data-driven models that are trained with RANS CFD simulations of the full vessel at specified conditions to sample the parameter space that the vessel operates in during a maneuver. The goal is for the data-driven model to determine the force vector of the propeller and rudder based upon the instantaneous forward speed, sway velocity, rudder angle, and yaw rate of the vessel.

By utilizing the data-driven model for the propeller and the rudder, only the bare hull is needed for a RANS CFD maneuvering calculation which significantly reduces the computational cost. This reduced computational cost can allow engineers to perform more simulations to gain a better understanding of how a vessel performs in various maneuvers in different conditions which will enable the design of safer vessels. The method is demonstrated for RANS CFD maneuvering calculations but is also applicable for problems such as autonomous ship control and optimal vessel route planning based on sea conditions.

1.1 Literature Review for Modeling a Propeller for Maneuvering and Seakeeping

The maneuvering and seakeeping characteristics of a vessel can be modeled in several ways including experiments, analytical methods, semi-empirical methods, and CFD. Experiments are useful for examining model scale vessels but require a physical model and time in a tow tank and the results must be scaled to determine full-scale vessel performance. CFD can be used to analyze both model scale and full scale vessels (*Tezdogan et al.* (2015)). CFD also can resolve viscous effects and provide more details than less expensive methods. CFD methods like RANS Volume of Fluid (VOF) CFD can be used to capture free surface and viscous flow effects but are computationally expensive. *Araki et al.* (2012) show how CFD can be more accurate

than a traditional systems-based model and also illustrate how a systems-based model can be improved by calibrating the coefficients with data from a select number of CFD maneuvering simulations.

Systems-based methods solve the equations of motion using a mathematical model, but require semi-empirical coefficients from either CFD or experiments to determine the propeller and rudder forces as shown by *Araki et al.* (2012). Systems-based methods are inexpensive and are widely used to examine vessel seakeeping and maneuvering characteristics as illustrated by *Umeda et al.* (2008), *Guo and Zou* (2017), and *Araki et al.* (2012); however, these methods can suffer from limited accuracy for arbitrary motion.

CFD maneuvering simulations of surface vessels typically use the Volume of Fluid (VOF) (*Hirt and Nichols* (1981)) formulation to account for the interface between the water and the air. A challenge with modeling the maneuvering or seakeeping of a vessel with CFD is to find a balance between the computational cost and accuracy. The propeller rotates at a high rate and is also much smaller than the hull. The difference in time and length scales between the propeller and hull drives up the computational cost since the vessel must be modeled with small temporal discretization and small spatial discretization around the propeller to accurately model the propeller forces.

In a maneuver the propeller spins about the shaft axis and the rudder also rotates about the rudder stock causing the vessel to turn. The rotating mesh for the propeller and/or rudder can be connected to the outer mesh via the overset method (*Carrica et al.* (2012), *Wang et al.* (2018)) or the sliding mesh method *Mizzi et al.* (2017). A challenge with the rotating sliding mesh approach is that due to the close proximity between the rudder and propeller these two zones may overlap. If the zones overlap then this approach cannot be used unless a simplification to the geometry. Additionally, if the rudder has a stationary part it may need to be simplified to allow a rotating sliding mesh approach (*Piro et al.* (2020)), whereby the rudder geometry

is modified which also alters the performance. The overset grid approach can be used to model the rotating propeller and/or rudder as performed by *Araki et al.* (2012), *Wang et al.* (2018), and *Shen et al.* (2015). Implementing the overset grid adds a layer of complexity to the model setup but can effectively handle multiple moving and stationary surfaces that are in close proximity to each other, but the computation expense is still large.

Both the rotating sliding mesh method and overset grid method are computationally expensive since the transient solution must be calculated for both the propeller and the vessel. When the propeller is discretized and spinning, the simulation must take a very small time step to accurately resolve the propeller forces, however, discretizing the propeller can provide accurate propeller forces which are critical for modeling vessel seakeeping and maneuvering performance. *Carrica et al.* (2012) performed a CFD study on the Office of Naval Research (ONR) Tumblehome with rotating propellers and rotating rudders to examine broaching. In 2015, a model scale simulation of a 10/10 zigzag maneuver of the KRISO containership at model scale took 123 hours with 96 E5-2650 v2, 2.60 GHz processors on the Neon HPC cluster at the University of Iowa. The total grid size was 3.8 million cells and the Froude number was 0.26 (*Shen et al.* (2015)). More recently, in 2018, a 10/10 zigzag in calm water of the ONR Tumblehome operating at a Froude number of 0.2 at model scale cost 325 hours of clock time with 40 processors to compute 19.5 seconds of simulated time (*Wang et al.* (2018)).

To mitigate the computational cost in a CFD analysis a body force propeller model can be used which enables a larger time step. A body force propeller model applies the force of the propeller to the equations of motion and to the fluid domain using a body force term in the Navier-Stokes equations. Similarly, for systems-based maneuvering calculations the force of the propeller is determined using a propeller force model. A common method for determining the propeller force is to only determine the thrust,

and in some cases the swirl, of the propeller based upon the open water propeller performance examples of which are shown by *Araki et al.* (2012), *Hoekstra* (2006), *Mousaviraad* (2010), *Sadat-Hosseini et al.* (2014), *Jin et al.* (2019), and *White et al.* (2019). The thrust of the propeller operating behind the vessel can be determined from the open water thrust coefficient, the propeller revolution rate, the propeller diameter, the wake fraction, the vessel velocity, and the density of water as performed by *Araki et al.* (2012).

The propeller forces can also be determined from numerical methods like panel methods or Boundary Element Methods (BEM) (*Chu et al.* (2018), *Gaggero et al.* (2019b)), lifting-line or lifting-surface methods (*Mao and Young* (2016), *Parsons et al.* (1980)), Vortex Lattice Methods (VLM) (*Cura-Hochbaum* (2006), *He and Kinnas* (2017), *Kinnas et al.* (2012)), Blade Element Momentum Theory (BEMT) (*Dubbioso et al.* (2017), *Gaggero et al.* (2019b), *Ortolani et al.* (2018), *Trodden et al.* (2016), *Winden* (2014)), and CFD based methods(*Knight and Maki* (2019), *Yao* (2015)). BEMs account for asymmetric inflow to the propeller due to the interaction with the hull and rudder, and can also account for geometric effects like skew. Other potential-flow methods like lifting-line methods or VLM can be used to determine the sectional lift and distribution along the propeller blade. BEMT combines blade element theory (*Froude* (1878)) with momentum theory (*Froude* (1889), *Rankine* (1865)). BEMT accounts for the induced velocity of the propeller and the variation in inflow to the propeller disk due to interactions with the hull and the rudder.

Potential-flow methods can be less accurate than viscous-flow methods like CFD when the propeller operates off design. *Gaggero et al.* (2019b) examined off design behind condition propeller loads between the RANS CFD prediction of a discretized rotating propeller, a BEM implementation and a BEMT implementation. The findings illustrated that the BEM body force distribution agreed better with the RANS CFD prediction but also noted that BEMT implementation in a maneuvering CFD

package is simpler. *Cura-Hochbaum* (2006) computed the results for a propeller with different inflow velocities and implemented a body force method in a RANS calculation of a maneuvering vessel in which the force for each region of the disk are determined from offline calculations and the local inflow velocity to the propeller disk. *Yao* (2015) developed a database propeller force model that is based off of RANS CFD simulations of an open water propeller with a sliding grid at different advance coefficients and oblique flow angles.

Vessels operate in the ocean environment and encounter irregular waves of different heights and frequencies. Unsteady propeller forces can be determined as a function of the velocity and the acceleration of the propeller (*Parsons and Vorus* (1981), *Martio et al.* (2017), *Knight and Maki* (2019), *Li et al.* (2018)). However, to predict the propeller forces for a vessel performing a maneuver the acceleration is typically low enough that forces can be determined with quasi-steady methods that neglect the acceleration of the propeller. In general, the acceleration of the propeller is neglected and only the velocity at the propeller plane is used for a vessel seakeeping or maneuvering simulation when using a body force propeller model.

Beyond the calculation of the propeller force, the propeller force distribution is also important since the rudder often operates in the wake of the propeller. Depending upon the body force distribution the forces on the rudder can change. There are several methods that are used to distribute the body force. The simplest approach is to apply only the thrust in a uniform distribution over the body force zone which encompasses the swept volume of the propeller. The thrust distribution can be modified to be more representative of an open water propeller and to include the effects of the torque on the flow by using models like those developed by *Hoekstra* (2006) and *Hough and Ordway* (1965). These methods are both developed for open water propellers and the force distribution neglects the effect of the interaction of either the hull or the rudder on the propeller force distribution; but the model developed by

Hoekstra includes the effect of a duct for a ducted propeller. *Sadat-Hosseini et al.* (2014) compared the effect of three propeller modeling techniques for a maneuvering vessel. This included two body force distributions including an axisymmetric body force distribution and the Yamazaki potential-flow propeller model (*Yamazaki* (1968)) as well as a discretized rotating propeller case. *Sadat-Hosseini et al.* (2014) showed that the Yamazaki propeller model led to maneuvering characteristics that were in better agreement with discretized propeller than the axisymmetric propeller model. *Phillips et al.* (2010) examined an open water propeller with a rudder in its wake using three body force propeller models including a uniform thrust approach, a Hough and Ordway distribution, and a BEMT distribution. This study found that the uniform thrust case, which neglects swirl, does not predict the drag of the rudder as well as the other two methods. This study also notes that the BEMT approach is the best approach of the three methods examined since it can account for the effect of the hull and the effect of the rudder acting on the propeller. *Broglia et al.* (2013) also compares BEMT to a modified Hough and Ordway distribution that accounts for the velocity at the propeller plane due to hull effects and also determines the side force of the propeller.

In a maneuver, the thrust is not the only propeller force that is critical to properly determining the vessel motions. The propeller side force can also be important (*Broglia et al.* (2013), *Guo et al.* (2018)). The propeller side force applies a force in the sway degree of freedom of the vessel and also applies a yaw moment. Different methods can be used to determine the propeller side force. *Guo et al.* (2018) analyzes the ONR Tumblehome with a systems-based method and calculates the propeller side force based upon the hydrodynamic derivative of the propeller side force, the drift angle, and the yaw rate. Other methods can be used to determine the propeller side force such as BEMT (*Broglia et al.* (2013), *Dubbioso et al.* (2017)) or it can be determined directly with a discretized propeller. Systems-based methods that use

hydrodynamic derivatives for the determination of the propeller side force neglect the variation of the propeller side due to changes in forward speed, non-linear effects, and cross-coupling between forward speed and the effective inflow angle. *Dubbioso et al.* (2017) investigated the multiple degree of freedom propeller force and distribution using BEMT coupled with RANS CFD and the effect of the rudder deflection angle. The work of *Dubbioso et al.* (2017) also examines simpler acutator disk methods that can predict the propeller side force like that of *Ribner* (1943). *Dubbioso et al.* (2017) shows how Ribner's method can provide reasonable predictions especially for low rudder angles.

An effective propeller model is inexpensive to use in a maneuvering simulation and should accurately predict the multi-degree of freedom force of the propeller operating in the behind condition. Additionally, the propeller model should either enable accurate prediction of the rudder force via an accurate body force distribution or allow for an accurate rudder force prediction through another means. One way of accomplishing this is by using a data-driven model trained with high fidelity simulations of a discretized propeller operating in the behind condition with a deflected rudder to account for both hull and rudder interaction effects. The ultimate goal when using data-driven methods is to be able to predict the quantity of interest as accurately as the method which is used to train the data-driven model. Additionally, if a data-driven model is used to predict the force of a propeller for use in a maneuvering simulation, it is desirable to limit the amount of training required since high fidelity simulations of the the propeller are expensive.

1.2 Literature Review for Data-Driven Modeling

Data-driven methods and machine-learning techniques use data to train mathematical algorithms to understand a phenomenon and to predict a result based upon new data. There are many data-driven techniques, some of which include linear

regression, neural networks, and Gaussian Process Regression (GPR) or Kriging.

Neural networks are most applicable for multi-dimensional problems and with large amounts of training data. *Hornik et al.* (1989) show that any mathematical function can be predicted to any level of accuracy by using a neural network with sufficient training and a sufficient number of hidden units. The application of neural networks in the discipline of fluid mechanics is broad. *Singh et al.* (2017) uses a neural network to improve a RANS turbulence model to better predict separation over an airfoil. *Zhang and Duraisamy* (2015) compares neural networks to Gaussian process regression for the purpose of data-driven turbulence modeling. *Abramowski* (2005) applies neural networks for the prediction of a propeller operating in crash-back, crash-ahead, and backing conditions. *Roddy et al.* (2008) uses a feedforward neural network to predict the multi-degree of freedom propeller force for a maneuvering submarine. The neural network is trained with a series of free-running maneuvers of the submarine including operational parameters in the crashback flow regime. *Roddy et al.* (2008) uses data from 65 total maneuvers in the study, wherein 53 maneuvers were used to train the model and 12 were used to validate the model. *Knight and Maki* (2020) compare the performance of a neural network to regression for the determination of the propeller thrust, torque, and side force for a propeller operating in both open water and in the behind condition for a container ship without a rudder present. In this study the two degree of freedom effect of oblique flow angle and the advance coefficient are considered such that only two degrees of freedom are considered unlike *Roddy et al.* (2008) who considered a full six degree of freedom model for a submarine. To train the models, *Knight and Maki* (2020) uses a significantly smaller amount of training data than *Roddy et al.* (2008) to examine a lower dimensional space for which the CFD results of one unsteady maneuver as well as open water propeller data extended to the behind condition using the thrust identity *ITTC* (2017). *Vesting and Bensow* (2014) found that surrogate models constructed with a

feed forward neural network and with a Kriging model could be used to optimize a propeller. One drawback of neural networks can be their sensitivity to over-training which can especially be a problem for noisy data, especially when data is limited; however, procedures like early stopping can mitigate this *Prechelt* (1998).

Kriging was originally developed for geostatistics (*Krige* (1951), *Matheron* (1963)), but it is also commonly used to generate surrogate models for engineering applications as originated by *Sacks et al.* (1989). Response surfaces generated with Kriging are popular for optimization studies since it is desirable to evaluate the function in a computationally efficient manner and for gradient based optimization the objective function should be a continuous function (*Bouhlel and Martins* (2019), *Liem et al.* (2015), *Vesting and Bensow* (2014)). *Laurenceau and Sagaut* (2008) compared different sampling techniques for modeling an aerodynamic response surface with Kriging and Cokriging which also uses the gradient at each sample point to improve the response surface. This study illustrates that a benefit of Kriging is that since both a prediction and a Kriging model uncertainty is provided throughout the design space, samples can be selected to construct an accurate response surface more efficiently. *Filip et al.* (2014) used CFD to model nine different bulb designs for a container ship and developed a Kriging response surface that was used to develop an improved design.

There are many variations of Kriging based methods with different underlying assumptions. Gaussian Process Regression (*Rasmussen and Williams* (2006)) is a useful tool and in general assumes underlying noise in the samples. This can be useful for experiments when the underlying variance is known *Zhang and Duraisamy* (2015) and is applicable for applications like vessel maneuvering identification (*Xue et al.* (2020)). *Gaggero et al.* (2019a) examined multi-fidelity Gaussian Process Regression applied to an open-water propeller in straight ahead motion.

Body force propeller models that use the open water propeller curves often describe

the thrust coefficient as a function of the advance coefficient which can be determined by linear regression. Additionally, systems-based methods often use linear regression to determine the propeller side force (*Guo et al.* (2018)). Other regression models like multiple regression are used in the literature. *Bernitsas et al.* (1981) developed a regression form for the open water thrust, torque, and efficiency of the Wageningen B-Series propellers based upon the advance coefficient, the pitch to diameter ratio, the expanded area ratio, and the number of blades. *Bhattacharyya et al.* (2016) developed a multiple linear regression based model to determine the forces on both the duct and the propeller for a ducted propeller to improve the prediction of the performance based on scaling. To train the model, 75 simulations were used to account for differences in scale, advance coefficient, and pitch ratio. In this study the simulations are run in a structured form such that five variations in the scale, five variations in advance coefficient, and three variations in pitch ratio are examined.

Various methods can be used to sample the parameter space to train and validate a data-driven model. For the purpose of developing a data-driven model to determine the propeller force during a maneuver it is necessary to have an accurate model, one that is inexpensive to use during the simulation, but also one that has a reasonable training expense. A full factorial grid approach uniformly samples the parameter space in uniform increments. Alternative approaches like Latin Hypercube Sampling (LHS) are more suitable for higher dimensions and are a popular way for sampling a design space (*Laurenceau and Sagaut* (2008), *Tang et al.* (2017)). LHS points are stratified and in the simple case of two dimensions only one sample point is used for each row and column. Alternatively, purely random sampling could be used, but this may neglect regions of the parameter space.

When developing data-driven models it is important to characterize the uncertainty associated with the model. The goal is to accurately calculate the truth with a predictive model. Both experimental and computational calculations contain underly-

ing error ϵ_d. Depending upon the underlying approach used to perform an experiment or computation the uncertainty in the calculation can be small or large. When a data-driven model is constructed with training data and validated with validation data, the model can only be expected to be as accurate as the data that it is trained with. Thus, the uncertainty in the data used to train and validate a data-driven model is propagated into the model.

In addition to the underlying uncertainty of the data used to train and validate the model, there is error in the prediction of the data-driven model as well. In machine learning, the data for developing the data-driven model is generally split into a training cohort and a validation cohort. A data-driven model is constructed with the training data, and to limit over-training to the data that it is trained with, it is validated with the validation data. The goal for a data-driven model is for the model to predict seen and unseen (validation) data as accurately as the method that is used to train the model. The data-driven modeling error with respect to the data that is used for training and validation can be referred to as ϵ_r.

Regardless of the method used to sample and develop a data-driven model, there is error in the model ϵ_m. ϵ_m is a measure of how close the model prediction is to the truth. The ϵ_m is a function of the error inherent in the data that it is trained with ϵ_d and the error of the data-driven model's prediction of the data that it is trained and validated with ϵ_r.

In the case of high-fidelity data for which the underlying uncertainty in the data can be assumed to be negligible, the ϵ_m is the same as the ϵ_r; however, for data-driven models that are constructed for complex problems, the underlying data often contains some degree of uncertainty. In the case of uncertain training data, the model is trained with data that contains error and is validated against data that contains error. The ϵ_m can be quantified by comparing the prediction of the data-driven model relative to the best available calculation of the truth. For CFD predictions, often the

finest grid or experimental results are treated as the truth.

1.3 Outline and Contributions of this Thesis

This work investigates a novel approach for modeling the control surfaces of a vessel using CFD simulations to generate data-driven models for use in maneuvering and seakeeping analyses. The objective of this work is to demonstrate a framework that allows for the use of a data-driven model to predict the forces on the vessel's propeller(s) and rudder(s), such that the forces can be applied during a maneuvering simulation. The goals for the data-driven model are to be efficient to train, to accurately predict the propeller and rudder forces with regards to the method used to train the model, and to significantly reduce the computational cost of a maneuvering simulation.

A novel approach is used to train and validate data-driven models based upon CFD simulations of the propeller and rudder operating in the behind condition. This work applies a unique methodology to efficiently develop a propeller model that captures the effects of the complex viscous flow interaction between the hull, propeller, and rudder in a maneuver by training the data-driven model from CFD simulations of the propeller operating in the behind condition with a deflected rudder. In the literature, propeller force models do not directly account for the viscous effects of the discretized rotating propeller and rudder. The data-driven models presented in this work are trained with CFD simulations that directly model the propeller and rudder in the behind condition to account for the complex viscous flow effects. Furthermore, the data-driven models constructed in this work remove the need for complex computational techniques like overset grids since only the bare hull is required for maneuvering simulations and the forces of the propeller(s) and rudder(s) are calculated using the data-driven model. The approach outlined in this work significantly reduces the computational cost of performing CFD maneuvering simulations. Additionally,

the training simulation cost for the data-driven models is significantly less than the cost required to perform just one maneuvering simulation with a discretized propeller and rudder. The model training expense is justified even if only one maneuvering simulation is performed, but the benefits of the model are especially advantageous when many maneuvering simulations are performed.

Different data-driven methods and machine learning techniques are examined to train and validate the propeller and rudder model. Each set of data contains a different level of uncertainty based upon discretization error and turbulence modeling error. The sources of error are quantified and the effect on the predictions of the data-driven model are analyzed. This work provides a comprehensive analysis of how much training data is required to train the model and how accurate the data-driven model is. To accomplish this, different grids and different turbulence models are used in the training CFD simulations and the results are compared. The accuracy of the data-driven models is calculated from the underlying uncertainty in the data used to train and validate the model as well as the error in the data-driven model prediction. A comprehensive analysis of the effect of data-driven model uncertainty and its effect on the maneuvering characteristics of vessels is explored in detail.

Specifically, this work investigates the determination of the propeller and rudder force coefficients using both linear regression and Gaussian process regression. The amount of training data relative to accuracy is also investigated for both methods. Linear regression is a technique that has been used for many years, and it is applied in a novel way to determine the forces of the propeller and the rudder through the lens of machine learning validation techniques. Similarly, Gaussian process regression is a potentially more robust data-driven technique that is suitable for generating surrogate models. The accuracy of each of the models are evaluated in the context of the accuracy and the volume of training data available. The implementation of the body force propeller and rudder model reduces the computational cost of the

maneuvering simulations, but training is also limited such that the cost of training the model is less than the cost of one maneuvering simulation.

This study applies the method of using a data-driven propeller model to two vessels. The KRISO Container ship (KCS), which is a single screw container ship with a semi-horned rudder, is analyzed at model scale. The seakeeping and maneuvering characteristics of this vessel have been widely studied in the literature (*Piro et al.* (2020), *White* (2020), *Shen et al.* (2015), *Wang et al.* (2018), *SIMMAN* (2008), *SIMMAN* (2020)). The ONR Tumblehome is also examined. The ONR Tumblehome is a twin screw vessel with two rudders and is also analyzed at model scale. The maneuvering characteristics of the ONR Tumblehome have been analyzed using a variety of methods in the literature ranging from experimental results (*Elshiekh* (2014), *Sanada et al.* (2019), *SIMMAN* (2020)), to systems-based methods (*Araki et al.* (2012), *Guo et al.* (2018)) to full RANS CFD simulations with a discretized propeller (*Carrica et al.* (2012), *Wang et al.* (2016)).

Practitioners of methods described in this work can elect to use the method of their choice to train the data-driven model. In this work RANS CFD is used for developing and evaluating the model. For example, for off-design conditions very high fidelity simulation tools like LES or DES could be used; while for on-design conditions a lower fidelity tool like BEM or BEMT could be used. The goal of this work is to illustrate how a data-driven model for the propeller and rudder can be trained at a reasonable cost, implemented into a custom CFD package with reasonable effort, and accurately predict the forces of the propeller and rudder for very low computational cost while maintaining accuracy of the viscous CFD solution. In this work the models are applied to CFD simulations; however, this method could also be used in conjunction with a vessel's autopilot to improve the performance of automated vessel control. This method can also be used to provide a propeller and or rudder model for a systems-based maneuvering method.

This work expounds upon the author's published works that form the foundation of this dissertation (*Knight and Maki* (2019), *Knight and Maki* (2020), *Knight et al.* (2020), and *Piro et al.* (2020)). Chapter II describes the methods used in this work and provides a mathematical foundation for this work. Chapter III develops the data-driven propeller and rudder model for the KCS, while Chapter IV applies the data-driven propeller model for the analysis of the turning circle characteristics of the KCS both in calm water and in waves. In Chapter V the data-driven propeller and rudder model is developed for application to the twin-screw twin-rudder ONR Tumblehome and in Chapter VI the turning circle characteristics of the ONR Tumblehome are examined in calm water and in regular waves.

CHAPTER II

Framework for Data-Driven Propeller and Rudder Model

The force generated by the propeller depends upon the motions of the vessel as well as the complex flow interaction between the hull, propeller, and rudder. The propeller and rudder operate in the wake of the hull. The propeller pushes the vessel forward, but also has forces acting in other degrees of freedom that can affect the maneuvering characteristics of the vessel. Additionally, the rudder operates in the propeller wake and this has a significant effect on the maneuvering characteristics of the vessel. During a maneuver the propeller and rudder operate off-design which can lead to flow separation and stall. The propeller and rudder move with the rigid ship hull and also have their own respective rotations relative to the hull. The rudder rotates about the rudder stock to turn the vessel and the propeller rotates about the propeller shaft which pushes the vessel forward.

There is a significant difference in time and length scales in the ship maneuvering problem. For a full scale vessel, a maneuver like a turning circle or a zig-zag takes on the order of minutes, the wave period for ambient waves is on the order of seconds to tens of seconds, and the propeller blade passing frequency is a fraction of a second. The goal of this work is to derive data-driven propeller and rudder models to reduce the computational cost of ship maneuvering simulations using CFD, such that the

time-step required is driven by the time scale of the hull instead of the propeller.

This work removes the need to discretize the propeller and the rudder by developing a data-driven model to predict the force vector on each propeller and each rudder. In this work, CFD simulations are used to train and evaluate data-driven propeller and rudder models. Additionally, maneuvering simulations are performed to evaluate the maneuvering characteristics of different vessels. OpenFOAM version 2.4.x is used for all of the CFD in this work. For the maneuvering simulations, a customized six degree of freedom solver is used (*White* (2020), *White et al.* (2021)).

In this chapter, the reference frames and motions for the nonlinear maneuvering problem are presented first. Following a discussion on the ship maneuvering problem, a discussion on the determination of the propeller and rudder forces for arbitrary motion is presented. Following the general discussion of how to determine the propeller and rudder forces, the data-driven techniques used to determine the force coefficients are discussed. Finally, a simple case study comparing two data-driven techniques is presented for the simple problem of the open water thrust coefficient as a function of the advance coefficient for a propeller in open water.

2.1 Ship Maneuvering Reference Frames and Motions

When a vessel performs a maneuver there is a six degree of freedom force acting on the the hull, the propeller(s), and rudder(s). The nonlinear equations of motion are solved (*White* (2020), *White et al.* (2021)) to determine the motions and maneuvering characteristics of the vessel. In this work, when the maneuvering characteristics of the vessel are analyzed the hull is modeled with VOF CFD, but a data-driven model is derived to determine the multi-degree of freedom force acting on the propeller(s) and rudder(s).

The motions and coordinate systems of a vessel are illustrated by Fig. 2.1. To note, the vessel shown in this image is the KCS. There are two coordinate systems:

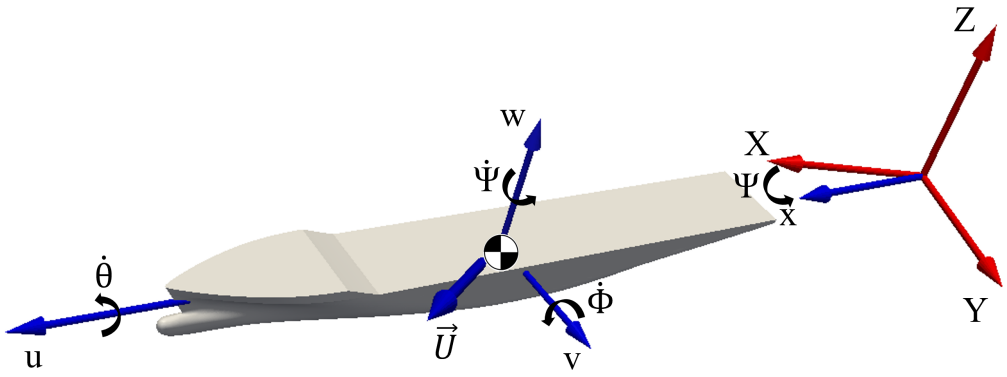

Figure 2.1: Motions and reference frames for the maneuvering problem .

the earth-fixed frame and the body-fixed frame. The earth-fixed coordinate system is specified at the original origin of the domain with X in the original forward direction of the vessel, Y in the direction towards port, and Z vertically upwards. The body-fixed frame is located at the center of gravity of the vessel with direction x towards the bow, y towards port, and z upwards. The ship velocity vector $\overrightarrow{U}$ has components u, v, and w for directions x, y, and z respectively. The rotation rates are about the x, y, and z axes and thus the roll rate is $\dot{\theta}$, the pitch rate is $\dot{\phi}$, and the yaw rate is $\dot{\psi}$. Right hand rule convention is used for the different rotations and rotation rates.

The force and moment acting on the vessel $\overrightarrow{F_V}$ is a six degree of freedom force, which is applied at the center of gravity of the vessel. The six degree of freedom force of the hull $\overrightarrow{F_H}$, propeller(s) $\overrightarrow{F_P}$, and rudder(s) $\overrightarrow{F_R}$ determine the force acting on the vessel as shown by Eqn. 2.1. The nonlinear equations of motion are solved in the customized OpenFOAM six degree of freedom solver. Fig. 2.2 shows a schematic with the components of the different forces as well as some geometric details that affect the motions of the vessel. The geometric details shown are the distance that the propeller is aft of the center of gravity of the vessel d, the distance that the rudder stock is aft of the vessel center of gravity d_r, and the rudder angle about the rudder

stock δ with positive δ correlating to the right hand rule about the upwards pointing rudder stock axis. The six degree of freedom force of the total force acting at the center of gravity of the vessel is $\overrightarrow{F_V}$, with three force components and three moment components. The force components of $\overrightarrow{F_V}$ are F_{Vx}, F_{Vy}, and F_{Vz} in the body-fixed x, y, and z directions respectively; the three moment components of $\overrightarrow{F_V}$ are M_{Vx}, M_{Vy}, and M_{Vz} about the body-fixed x, y, and z directions respectively. Each component of the six degree of freedom force is the sum of the components of the force on the hull, the rudder, and the propeller applied to the center of gravity of the vessel as shown by Eqn. 2.2. To note, in a maneuver, the propeller and rudder moments calculated about the center of gravity of the vessel are what affect the maneuver and this is what is illustrated in Eqns. 2.3. Separately, the six degree of freedom force about the propeller shaft axis at the hub fixed plane as well as the six degree of freedom force of the rudder about the rudder stock can affect other aspects of design and analysis.

$$\overrightarrow{F_V} = \overrightarrow{F_H} + \overrightarrow{F_P} + \overrightarrow{F_R} \tag{2.1}$$

$$\begin{aligned} F_{Vx} &= F_{Hx} + F_x + F_{Rx} \\ F_{Vy} &= F_{Hy} + F_y + F_{Ry} \\ F_{Vz} &= F_{Hz} + F_z + F_{Rz} \end{aligned} \tag{2.2}$$

$$\begin{aligned} M_{Vx} &= M_{Hx} + M_{Px} + M_{Rx} \\ M_{Vy} &= M_{Hy} + M_{Py} + M_{Ry} \\ M_{Vz} &= M_{Hz} + M_{Pz} + M_{Rz} \end{aligned} \tag{2.3}$$

The equations of motion can be described in terms Newton's second law as well as Euler's equations as shown by Eqn. 2.4 (*White et al.* (2021)). The equations of motion depend upon the ship state defined by the six degree of freedom motion, the propeller rate n, and the rudder angle δ. The force acting on the body is a function

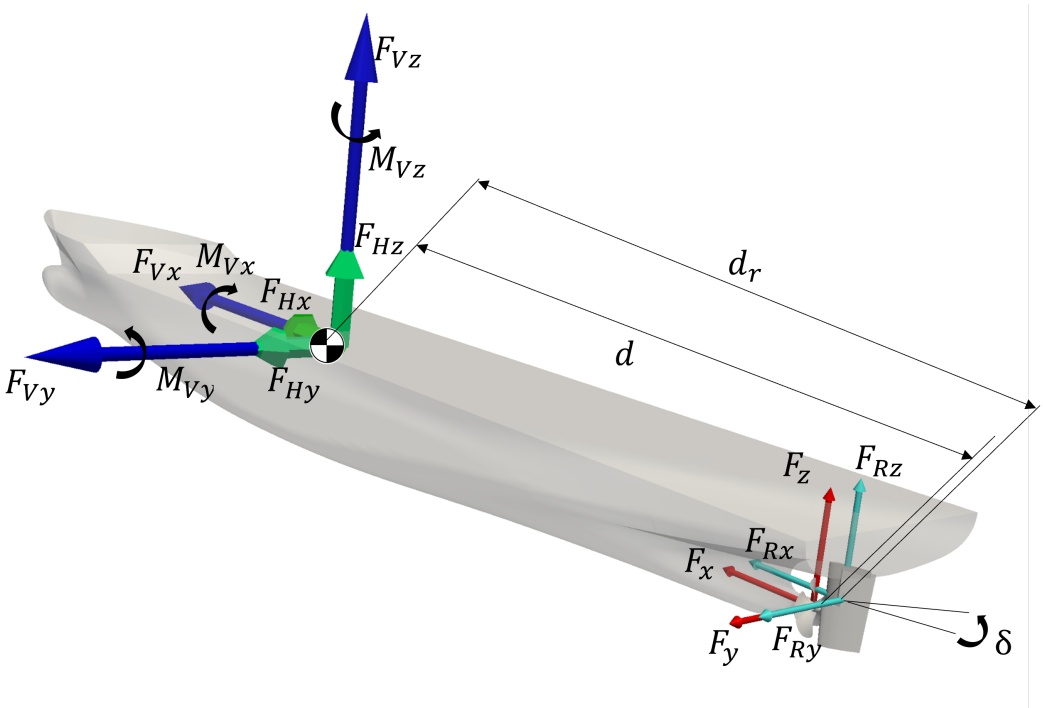

Figure 2.2: Forces applied to the vessel in the maneuvering problem.

of the force vector on the vessel projected into the earth-fixed frame $\overrightarrow{F_{VE}}$, the mass matrix of the vessel m, and the accelerations of the vessel in the earth-fixed frame $\ddot{X}$. Euler's equations of motion are a function of the applied moments $\overrightarrow{M_V}$, the rotational inertia matrix I, the rotational rate of the body about the vessel's center of gravity $\overrightarrow{\omega_V}$, and the rotational acceleration of the body about the vessel's center of gravity $\overrightarrow{\dot{\omega}_V}$

$$\begin{aligned} \overrightarrow{F_{VE}} &= m\overrightarrow{\ddot{X}} \\ \overrightarrow{M_V} &= I\overrightarrow{\dot{\omega}_V} + \overrightarrow{\omega_V} \times (I\overrightarrow{\omega_V}) \end{aligned} \tag{2.4}$$

The six degree of freedom propeller force is the force vector that describes the three forces and three moments. The three forces are: F_x (or thrust T when the propeller shaft axis is aligned with the body-fixed frame) aligned with the propeller

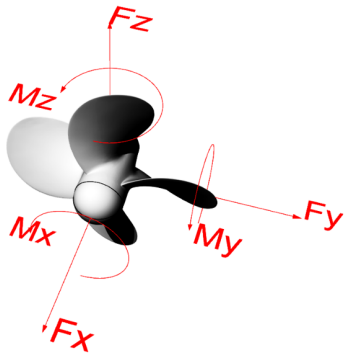

Figure 2.3: Forces acting on propeller.

shaft axis, F_y (or propeller side force S), and the vertical propeller force F_z. The thrust of the propeller T is aligned with the propeller shaft axis, so for a propeller with no shaft inclination F_x and T are the same. The moment contributions of the six degree of freedom propeller force applied to the center of gravity of the vessel are M_{Px}, M_{Py}, M_{Pz}. Additionally, the moments about the center of the propeller are denoted as: the moment about the x axis M_x, the moment about the y axis M_y, and the moment about the z axis M_z. The propeller torque Q is the moment required to drive the propeller about the propeller shaft axis. The torque induces a swirl effect to the flow which can affect the rudder force and it is also important for determining how much power is applied to the propeller. Fig. 2.3 shows a schematic for these forces for an open water propeller. Eqn. 2.5 shows the dimensionless form of the propeller force coefficients and Eqn. 2.6 shows the dimensionless torque coefficient, for which the other moments about the propeller can be calculated in the same form. The propeller force coefficients are determined by each respective force, the density of the fluid ρ, the propeller revolution rate n, and the propeller diameter D.

$$
\begin{aligned}
K_T &= \frac{T}{\rho n^2 D^4} \\
K_X &= \frac{F_x}{\rho n^2 D^4} \\
K_Y &= \frac{F_y}{\rho n^2 D^4} \\
K_Z &= \frac{F_z}{\rho n^2 D^4}
\end{aligned}
\tag{2.5}
$$

$$
K_Q = \frac{Q}{\rho n^2 D^5} \tag{2.6}
$$

The force vector of the propeller affects the maneuvering and seakeeping characteristics of a vessel. The generalized propeller force depends upon the six degree of freedom motions of the vessel, the propeller revolution rate, as well as the effect of the deflected rudder. While the torque required to rotate the propeller affects how much power must be delivered to the propeller, if the propeller revolution rate is held constant the most important forces on the propeller in a maneuver are the thrust and the side force. The propeller thrust affects the forward velocity of the vessel u. The propeller side force contributes to the sway velocity v of the vessel and also induces a yaw moment on the vessel which contributes to the yaw rate of the vessel $\dot{\psi}$. For a maneuver, the six degree of freedom force about the propeller hub fixed frame is not the only critical component, but the moment caused by the propeller about the center of gravity of the vessel is also important for the equations of motion. In turn, the force of the propeller can be determined as a function of the motions of the vessel specifically u, v, and the tangential velocity at the propeller plane determined by the product of $\dot{\psi}$ and the distance between the center of gravity of the vessel and the propeller plane d. Likewise, the rudder force is a function of u, v, $\dot{\psi}$, and the distance between the center of gravity of the vessel and the rudder stock d_r. The rudder force

is also a function of the thrust induced by the propeller upstream of the rudder. In this work, these motions are used to determine the propeller and rudder forces.

The most important propeller and rudder forces to consider for a maneuvering vessel are the forces in the x and y directions as well as the yaw moment about the z axis. Therefore, instead of determining the full six degree of freedom force on each propeller and rudder, the problem can be simplified to only determining the three dimensional force on each propeller and rudder. Certainly, other degrees of freedom can be calculated as well and included in the model. For example, if experimental results provide additional force parameters like propeller torque, it can be useful to include in the model to help validate it.

2.2 Modeling Propeller Forces for Arbitrary Motion

In this section, the equations for modeling the propeller with arbitrary vessel motion are discussed. When a vessel performs a maneuver, the effects of the oblique flow angle change the propeller and rudder forces. The propeller thrust affects how fast the vessel travels and the propeller torque affects how much power must be delivered. Furthermore, the propeller side force can be important to consider in order to accurately determine the vessel maneuvering motions. This work uses data-driven modeling to train a model to predict the propeller force, and the model is implemented in a maneuvering CFD simulation.

For straight ahead motion the forces of importance for a propeller are the thrust T and the torque about the propeller shaft Q. For a given propeller design, the force on the propeller is a function of the forward velocity u, the propeller revolution rate n, and the diameter of the propeller D. A propeller is easiest to model at its design condition, but in a maneuver the propeller operates off-design. When a vessel maneuvers it no longer operates in the surge-only degree of freedom. In addition to surge the other degrees of freedom are sway, heave, pitch, yaw, and roll. The sway and

yaw degrees of freedom produce an oblique flow angle acting on the propeller which can alter the forces on the propeller as well as the rudder. When the propeller operates off-design more separation occurs and higher fidelity modeling methods that are more expensive are required to achieve accurate predictions. Methods like Vortex Lattice Method (VLM), Boundary Element Models (BEM), or Blade Element Momentum Theory (BEMT) are most accurate near the design condition and are comparatively inexpensive methods to determine the propeller force. Viscous numerical methods like RANS CFD or LES can be more accurate off design but are also much more expensive. RANS CFD with wall functions is used in this work.

The dimensionless forward velocity that is used to model a propeller is the advance coefficient J, which is a function of u, n, and D as shown by Eqn. 2.7. In the behind condition the interaction between the hull and the propeller is important. The wake of the hull induces nonuniform velocity in the propeller plane. A common method to address this is to use the wake fraction w_f which is calculated as a function of the forward velocity of the vessel u and the advance velocity u_A as shown by Eqn. 2.8, where u_A is the average nominal velocity at the propeller plane. w_f indicates how much the flow is blocked, where a zero value of w_f indicates that the advance velocity is the same as if the propeller was operating in open water on average, whereas larger values of w_f indicate more of a velocity reduction at the propeller plane due to the wake of the hull. The wake fraction can be calculated with a variety methods ranging from averaging the velocity at the propeller plane with no propeller action (averaging the nominal wake) to using the thrust identity (*ITTC* (2017)) by performing a single behind condition simulation to extend the open water propeller curves to the behind condition by using the advance velocity based advance coefficient J_A as shown by Eqn. 2.9. This approach can be accurate for characterizing the forces in straight ahead motion as has been illustrated in *Knight and Maki* (2019). For maneuvering vessels, the effect of the rudder and other rigid body motions of the vessel are important

for determining the propeller force. Furthermore, the wake fraction can change as a function of oblique flow, thus, it is desirable to account directly for the forces acting on the propeller and rudder as a function of the oblique flow angle while accounting for the interaction between the hull, propeller, and rudder. In this work, the advance coefficient J is calculated using the forward speed of the vessel in accordance with Eqn. 2.7. Thus, the forward speed of the vessel is used and the effect of the wake is ignored when calculating this parameter; the effect of the wake of the hull is captured by determining the force of the propeller operating in the behind condition for the training CFD simulations.

$$J = \frac{u}{nD} \tag{2.7}$$

$$w_f = \frac{u - u_A}{u} \tag{2.8}$$

$$J_A = \frac{u_A}{nD} \tag{2.9}$$

The drift angle β is the angle between the forward velocity and the sway velocity as shown by Eqn. 2.10. For a vessel that operates purely with forward velocity and sway velocity, the drift angle is representative of the oblique flow angle at both the propeller and rudder positions; however, during a maneuver, the effect of the yaw rate and the distance of the respective surface from the center of gravity of the vessel alters the oblique flow angle for each respective surface. The oblique flow angle for a propeller operating in the behind condition β is a function of the the forward velocity, the sway velocity, the yaw rate $\dot{\psi}$ (measured in radians/second) and the longitudinal distance between the propeller and the vessel's center of gravity d as shown by Eqn. 2.11. Similarly, the oblique flow for the rudder β_R is the oblique flow angle at the rudder stock which is calculated as a function of the rigid body motion of the vessel in terms

of u, v, $\dot{\psi}$, and the distance between the center of gravity of the vessel and the rudder stock.

$$\beta = \tan^{-1}\left(\frac{v}{u}\right) \tag{2.10}$$

$$\beta_P = \tan^{-1}\left(\frac{v - d\dot{\psi}}{u}\right) \tag{2.11}$$

$$\beta_R = \tan^{-1}\left(\frac{v - d_r\dot{\psi}}{u}\right) \tag{2.12}$$

When the propeller acts in the behind condition the wake of the hull changes and this also changes the inflow to the propeller. When the propeller is exposed to oblique flow the propeller produces a side force S (or F_y) that can be important to include for the accurate prediction of a vessel's maneuvering characteristics. The side force of the propeller not only applies a side force to the vessel, but it also produces a yaw moment about the center of gravity of the vessel. The oblique flow also leads to a vertical component of the force which can also be modeled, but in this work that term is neglected for the maneuvering simulations. The data-driven models derived in this work simplify the full six degree of freedom force and only require the calculation of K_X, K_S, and the yaw moment generated by the propeller. In one of the case studies performed in Chapters III and IV, the experiments with which the model are compared also predicted the torque, so the torque is also calculated for that case. Eqn. 2.13 shows the propeller force coefficients that are calculated in this work for application in the CFD based maneuvering simulations. To note, in the analysis of the KCS in Chapters III and IV, the shaft inclination is 0° so K_X and K_T are identical. The yaw moment induced by the propeller is shown by Eqn. 2.14 and is a function of F_X, S, d, and the distance that the propeller is off center d_y (with positive d_y towards port).

$$
\begin{aligned}
K_X &= \frac{F_x}{\rho n^2 D^4} \\
K_S &= \frac{S}{\rho n^2 D^4} \\
K_Q &= \frac{Q}{\rho n^2 D^5}
\end{aligned}
\tag{2.13}
$$

$$
M_{Pz} = -(d\ S + d_y\ F_x) \tag{2.14}
$$

2.3 Rudder Forces for Arbitrary Motion

In addition to the propeller force, the rudder force is also critical for determining the maneuvering characteristics of the vessel. The importance of modeling the propeller forces accurately is important not only because the propeller forces affect the vessel's equations of motion, but the propeller also affects the inflow to the rudder and thereby affects the rudder force. The rudder operates in the wake of the propeller and the induced velocity of the propeller plays an important role in the rudder force. When the vessel is maneuvering, the rudder is deflected and the rudder changes the inflow to the propeller, which alters the propeller force. Thus, the rudder force is a function of the full eight degree of freedom ship state vector. However, the most critical dimensional components of the state are u, v, $\dot{\psi}$, n, and δ. Thus, for a given δ, a data driven model can be constructed to determine the rudder force coefficients as a function of J and β if $\dot{\psi}$ is is neglected while training the model.

Alternatively, the rudder could be discretized if an accurate propeller body force distribution can be determined. Some approaches to develop a body force distribution include using potential flow distributions, BEMT, or a data-driven model to predict the distribution. All of these approaches inherently contain a level of error, so in this

work the force of the rudder is directly calculated with a similar data-driven approach as is used to model the propeller. This also alleviates the need to discretize the rudder in the maneuvering simulation and removes the need to determine the propeller body force distribution.

The most important rudder forces to account for in a maneuver are the forces in the x and y directions as well as the yaw moment contribution. The rudder forces can be non-dimensionalized in terms of the inflow to the rudder velocity u_R, the lateral area of the rudder S_R, the density of water, and the force components. The force component in the forward direction in the body-fixed frame is F_{Rx} and the corresponding force coefficient is C_{Fx} as shown by Eqn. 2.15. Similarly the force component in the lateral direction in the body-fixed frame is F_{Ry} and the corresponding force coefficient is C_{Fy} as shown by Eqn. 2.16. The rudder yaw moment about the center of gravity of the vessel M_{Rz} is critical for evaluating a vessel's maneuvering characteristics. The rudder yaw moment coefficient C_{Mz} can be determined using Eqn. 2.17 in terms of the inflow to the rudder velocity, the lateral area of the rudder, the density of water, the rudder yaw moment, and the magnitude of the distance between the center of gravity of the vessel and the rudder stock d_r. The C_{Mz} can also be determined by approximating it as the product of C_{Fy} and d_r as shown by Eqn. 2.18. In the case of a single rudder that is aligned with the center line of the vessel Eqn. 2.18 is a convenient approach. On the other hand, if the rudder is off-centerline, it may be easier to use Eqn. 2.17 since the moment induced is a function of both the side force of the rudder as well as a smaller component due the resistance of the rudder and the distance that the rudder is from centerline.

$$C_{Fx} = \frac{2F_{Rx}}{\rho u_R^2 S_R} \tag{2.15}$$

$$C_{Fy} = \frac{2F_{Ry}}{\rho u_R^2 S_R} \tag{2.16}$$

$$C_{Mz} = \frac{2M_{Rz}}{\rho u_R^2 S_R d_r} \tag{2.17}$$

$$C_{Mz} = -d_r C_{Fy} \tag{2.18}$$

The thrust coefficient C_T can be described as a function of K_T and J as shown by Eqn. 2.19 which in turn can be used to determine the axial velocity at the rudder u_{Rx} as shown by Eqn. 2.20. This approach is similar to that used by *Kose et al.* (1981) which is used for systems-based methods that determine the rudder force using empirical coefficients such as the work performed by *Araki et al.* (2012). The approach used here simplifies the empirical model and neglects the velocity reduction at the rudder due to the hull wake, but it incorporates the effect of the induced velocity of the propeller. The lateral velocity at the rudder u_{Ry} is determined as a function of the sway velocity of the vessel, the yaw rate of the vessel, and the distance between the rudder stock and the center of gravity of the vessel d_r as shown by Eqn. 2.21. The rudder velocity is the resultant magnitude of the lateral velocity and the axial velocity as shown by Eqn. 2.22.

$$C_T = \frac{8K_T}{\pi J^2} \tag{2.19}$$

$$u_{Rx} = u(1 + C_T) \tag{2.20}$$

$$u_{Ry} = v - d_r \dot{\psi} \tag{2.21}$$

$$u_R = \sqrt{u_{Rx}^2 + u_{Ry}^2} \tag{2.22}$$

In this work, only turning circle maneuvers are performed. Thus, the rudder force coefficients can be determined simply with the basis of the deflected rudder. On the other hand, if a zig-zag maneuver is performed, or another variation of a maneuver that requires the rudder to be deflected to different angles, small adjustments to the form of Eqn. 2.27 should be made. For example, depending upon which direction the rudder is deflected, it may be more convenient to use a β_R of $|\beta_R|$. For a single

screw vessel with one rudder, the performance for a port turn and a starboard turn will be different due to the effects of the propeller rotation direction with the rudder, for which the training sample space should include both positive and negative β. Additionally, to handle rudder angles that are not equal to the position that the rudder was trained at, interpolation could be used or the methods investigated here could be expanded to include a third feature of the rudder angle δ.

2.3.1 Modeling of Multiple Propellers and Rudders

When multiple rudders or propellers are present a separate data-driven model can be generated for each set. Depending upon the position of each respective propeller and rudder different interactions between each propeller, rudder, and hull occur. For example, for a twin screw vessel with two rudders operating in the wake of each respective propeller a regression model can be generated for the port side and starboard side propeller-rudder sets respectively. In this case, when the rudder angle is non-zero and/or there is a non-zero oblique flow angle the forces acting on the propellers and rudders are different on the port side and starboard side. The case study on the maneuvering of the ONR Tumblehome illustrates these effects. When the propellers and rudders are not located on centerline, the yaw moment contribution of each surface becomes a function of not only the side force and the distance that the surface is aft of the center of gravity of the vessel, but it also depends upon the force in the x direction generated by that surface and the distance that the surface is off of centerline.

Additionally, when there are multiple propellers and rudders the training for the propeller and rudder model does not depend upon the sign of δ. For example, in the case of a twin-screw twin-rudder vessel performing a zig-zag maneuver, if both rudders are deflected to the same angle and both propellers are either inward spinning or both are outward spinning, the data-driven model can be adapted for either positive or

negative δ. Based on the direction of β and δ that is used for training, if the direction of β and δ are switched, then the side that originally saw clean inflow (windward) now becomes the leeward side, and the side that originally was the leeward side becomes the windward side in the new model. When this occurs the side force of both the propeller and rudder flip in direction and the yaw moment also flips in direction.

2.4 Data-Driven Modeling Techniques

There are multiple data-driven algorithms that can be applied to propeller modeling such as linear regression, neural networks, and Gaussian process regression. In this work the focus is upon linear regression and Gaussian process regression. Each of these can be used in terms of the feature of the advance coefficient J and the oblique flow angle β.

Regardless of the data-driven model used, the application of the data-driven model to a maneuvering simulation is outlined in Fig. 2.4. The purpose of the data-driven model is to determine the propeller and rudder force coefficients based on the motions of the vessel. To accurately determine the propeller forces with a data-driven method a specific technique must be selected. Each data-driven method has benefits and limitations.

2.4.1 Training Data Generation

The force vector acting on the vessel is a six degree of freedom force. However, some of the forces from the propeller and rudder can be omitted while still accurately capturing the maneuvering characteristics of the vessel. This work uses behind condition RANS CFD simulations of the propeller with a deflected rudder to directly capture the effect of the viscous flow interaction between the hull, propeller, and rudder. A double-body approximation is applied such that the free surface effects are assumed small and are neglected, such that the Froude number does not have a

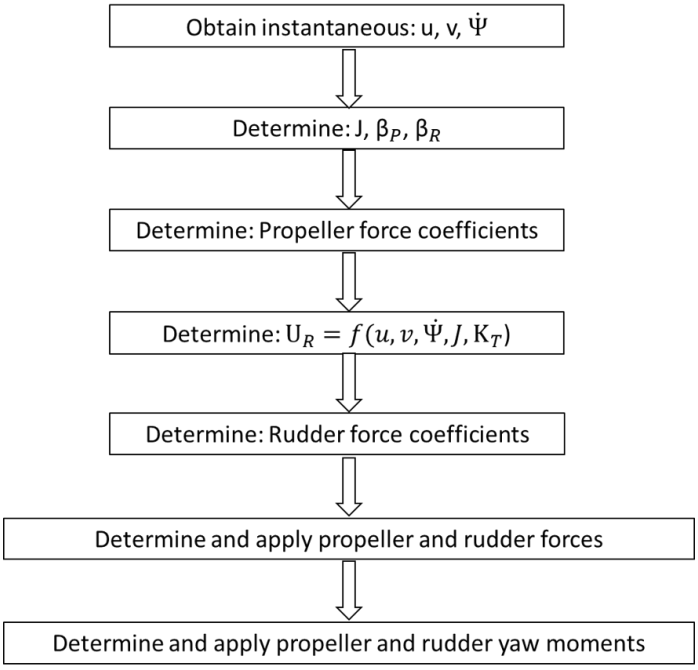

Figure 2.4: Flow chart for the implementation of the data-driven propeller and rudder model.

significant effect on the propeller and rudder performance. The method also assumes that the motion is slowly varying such that the force on the propeller and rudder depend only upon the instantaneous ship state.

The Froude number F_n, defined in Eqn. 2.23, is an important dimensionless parameter which relates the inertial and gravitational forces. The Froude number depends upon u, the length of waterline L_{wl}, and the gravitational constant g. This parameter is important for characterizing free surface effects like the wave drag on the hull, but it is assumed in this work that the propeller and rudder forces have small dependence upon the free surface effects. If the vessel operates with too large of a Froude number or if the vessel operates in a very high amplitude sea state, the double-body approximation which assumes a smooth free surface will break down.

$$F_n = \frac{u}{\sqrt{gL_{wl}}} \tag{2.23}$$

For the training simulations the vessel is held at a constant position, the rudder is fixed at the desired deflected angle, and the propellers rotate about the propeller shaft with a rotating sliding mesh. The forces of specific interest in this work are the thrust of the propeller, the torque of the propeller, the side force of the propeller, the yaw moment of the propeller induced at the center of gravity of the vessel, the resistance of the rudder, the side force of the rudder, and the yaw moment of the rudder induced at the center of gravity of the vessel.

Modeling a discretized propeller rotating in the behind condition using RANS CFD is expensive, therefore it is desirable to limit the number of data points required to train the data-driven models. In this work, Latin Hypercube Sampling is used to sample in the J-β space in the parameter range that the vessel operates in a maneuver. By sampling in the dimensionless J-β space instead of the dimensional space it allows for simple scaling to different vessel operating conditions. The J-β parameters are used to determine the dimensional velocity that is specified at the inlet of the CFD domain, such that the commensurate forward velocity and sway velocity that match the desired J-β values for each sample point. In the training CFD simulations, only forward velocity and sway are accounted for while n is held constant, thus, β, β_P, and β_R are all equal since there is no yaw. To note, in this work all of the models use the oblique flow angle specified in radians to train and utilize the models. Using the J-β parameter space reduces the number of independent variables of J, β, β_R, and β_P to just two independent variable (J and β). By using J-β space, the dependence upon u, v, $\dot{\psi}$, and n is accounted and the model can be scaled for n larger or smaller than it was trained with. Thus, by sampling and training a model to predict the force coefficients on the propeller and rudder in J-β space the dimensionalality of the problem is reduced and the forces can be scaled for a reasonable range of Reynolds number and provided that the Froude number is not so large that free surface effects alter the propeller and rudder force.

The Reynolds number, defined by Eqn. 2.24 is a dimensionless parameter which characterizes the importance of inertial versus viscous forces. The Reynolds number is a function of the characteristic velocity u_c, the characteristic length L_c, and the kinematic viscosity ν. In the laminar flow regime the Reynolds number is low and viscous forces are dominant. In the turbulent flow regime the Reynolds number is high and the inertial forces are dominant. Model scale propellers operate in the transitional flow regime between laminar to turbulent flow, whereas full-scale propellers operate in the turbulent flow regime. Different turbulence models assume different flow regimes. In this work the Spalart-Allmaras (*Spalart and Allmaras* (1994)) and $k-\omega$ SST (*Menter and Esch* (2001), *Menter et al.* (2003)) turbulence models are used which assume a fully turbulent boundary layer and wall functions are used. The use of wall functions can lead to less accurate results than using wall resolved grids with y^+ 1.0, however, they are less expensive (*Eca et al.* (2015)). A RANS based alternative to fully turbulent RANS models are transitional models which can account for the transition from laminar flow to turbulent flow (*Langtry and Menter* (2009), *Furst et al.* (2013)) and are applicable to model scale marine propellers (*Wang and Walters* (2012), *Baltazar and Rijpkema* (2017), *Webster et al.* (2019)), but the use of wall functions is limited with these models.

$$Re = \frac{u_c L_c}{\nu} \tag{2.24}$$

While Latin Hypercube Sampling is used in this work, other sampling techniques could be used as long as the sampled space is representative of the expected values of J and β that the vessel would operate in during a maneuver. To expand the coefficients that the different data-driven models determine in the J-β space to applications in a maneuver, the propeller oblique flow angle β_P and the rudder oblique flow angle β_R are used for the propeller and rudder forces respectively.

A data-driven model for a given application should be able to predict the output variable with a similar level of accuracy as the data that it is trained with. RANS CFD with the propeller discretized and rotating is used for training the data-driven model in this work, therefore, each data point analyzed to train the model is expensive. Restricting the number of required training and validation points is desirable to keep the overall expense of the model reasonable. Even with expensive methods like RANS CFD there is also error inherent to the training points used. The error of the data-driven model prediction with respect to experimental results is a function of the underlying error in the CFD based training and validation data (such as discretization error) as well as the error associated with the data-driven model itself.

An additional criterion for evaluation of different data-driven models is the ease of implementation. For engineers to incorporate this approach into their workflow they must be able to train the models and once a model is trained it must be implemented in a given CFD package. If two models have similar accuracy and training cost, it is desirable to implement the more straightforward algorithm.

2.4.1.1 Linear Regression

Linear regression is a simple but powerful tool for constructing and implementing a data-driven model. For a vessel undergoing sway and yaw, the oblique flow angle must be considered. A drawback of linear regression is that the form of the polynomial must be specified so the form of the response surface must be known or estimated. A sample feature vector which includes a scalar coefficient J to the first and second order, β_P to the first and second order, as well as a cross coupling term J-β_P is shown by Eqn. 2.25. The corresponding force coefficients K_X, K_S, and K_Q of the propeller are shown by Eqn. 2.26. In addition to the forces calculated from these force coefficients, the yaw moment contribution of the propeller is determined as a function of d, K_X, d_p and K_S as shown by Eqn. 2.14. The regression model determines the coefficients

based upon a set of training data for which the values of the force coefficients and the features are supplied. Alternative feature vectors could neglect the quadratic term, or additional terms could be added such as a cubic term. Additional features could lead to over-fitting the data depending upon how many samples are used to train the model; on the other hand, too few features could lead to an overly simplistic model. The linear regression models in this work are generated using the MATLAB function `fitlm`.

$$\vec{x} = [1, J, J^2, \beta_P, \beta_P^2, J\beta_P]^T \tag{2.25}$$

$$\begin{aligned} K_X &= a_1 + a_2 J + a_3 J^2 + a_4 \beta_P + a_5 \beta_P^2 + a_6 J \beta_P \\ K_S &= b_1 + b_2 J + b_3 J^2 + b_4 \beta_P + b_5 \beta_P^2 + b_6 J \beta_P \\ K_Q &= c_1 + c_2 J + c_3 J^2 + c_4 \beta_P + c_5 \beta_P^2 + c_6 J \beta_P \end{aligned} \tag{2.26}$$

To note, when the data-driven model is implemented in the CFD maneuvering simulation, the β_P and β_R are used in lieu of β since in a maneuver the propeller oblique flow angle is affected by the yaw rate and the distance that the propeller is aft of the center of gravity of the vessel. On the other hand, it is most convenient to train the model with only forward velocity and sway velocity such that β, β_P, and β_R are equal, and the J-β space only requires two dimensions to sample for each δ. If dimensional parameters were to be used and the yaw rate was accounted for then the sample space would become four dimensional (u, v, $\dot{\psi}$, n) for a given δ.

Using linear regression similar to Eqn. 2.26, the rudder force coefficients can be determined with Eqn. 2.27. It is important to once again note that for the training of the model, β and β_R are equal since there is no yaw rate, but in the course of a maneuver, β_R depends upon the yaw rate.

$$
\begin{aligned}
C_{Fx} &= e_1 + e_2 J + e_3 J^2 + e_4 \beta_R + e_5 \beta_R^2 + e_6 J \beta_R \\
C_{Fy} &= f_1 + f_2 J + f_3 J^2 + f_4 \beta_R + f_5 \beta_R^2 + f_6 J \beta_R \\
C_{Mz} &= g_1 + g_2 J + g_3 J^2 + g_4 \beta_R + g_5 \beta_R^2 + g_6 J \beta_R
\end{aligned}
\tag{2.27}
$$

The accuracy of the model can be evaluated based on how well it predicts the data that it is trained with, but more importantly how well it predicts unseen data. In this work, the regression model is trained and validated with RANS CFD simulations of the propeller operating in the behind condition with a rotating sliding mesh, therefore, it is desirable to also limit computational cost and mitigate the number of training and validation points. As noted in the literature review, many body force propeller models are based upon linear regression and determine only the thrust and the torque of the propeller based upon the advance coefficient of the vessel based upon the wake fraction. In this study the J is calculated as a function of the forward speed of the vessel and it is shown that the β can be important for the prediction of the propeller force.

2.4.1.2 Gaussian Process Regression

Gaussian Process Regression (GPR) is another data-driven technique that is suitable to generate a response surface. GPR is more complicated to implement in a custom CFD package for performing maneuvering vessel analysis, but has some distinct advantages over linear regression. GPR provides not only a prediction of the output variable based on the input parameters, but also provides an uncertainty bound, which can be useful for determining where to add additional samples in the parameter space to improve the accuracy. The uncertainty bound is available since the method uses a Gaussian process to develop the model. Additionally, if the input

data is noisy and the standard deviation of the noise is known, this can be incorporated into the model. For example, for CFD applications the discretization and/or turbulence modeling uncertainty could be considered; alternatively, for experiments, the experimental error can be used. If the input data is noise-free, then the error of the prediction at the training points is zero. In this work the GPR is modeled in MATLAB using the function `fitrgp` which is based off of *Rasmussen and Williams* (2006), which is subsequently summarized.

For each desired output variable y (for example K_T, K_S, or K_Q) the model is trained with N training points. Eqn. 2.28 shows that the output variable is determined by a function of the input vector $\vec{x}$, the weight vector $\vec{w}$, and the error between the function evaluation $f(\vec{x})$ and the output variable y denoted by ϵ, which is shown by Eqn. 2.29. The error ϵ is a Gaussian noise with a variance of σ_n^2 with a zero mean. If the samples used to train the model are assumed to be free of noise, then the model will predict the training point prediction with no error. If on the other hand, the level of noise in the sample points is known, this can be applied to the GPR. For the purpose of modeling the propeller forces, the input vector is the value of J and β for each correlating output parameter y. Thus, the set of training for K_T would be the vector of length N points of the CFD based calculation of K_T and the $\vec{x}$ would be the corresponding J and β values.

$$y = f(\vec{x}) + \epsilon = \vec{x}^T \vec{w} + \epsilon \tag{2.28}$$

$$\epsilon \sim \mathcal{N}(0, \sigma_n^2) \tag{2.29}$$

$f(\vec{x})$ can be described as a Gaussian process of the mean of the expected value of the function $m(\vec{x})$ and the covariance function $k(\vec{x}, \vec{x}')$, as shown by Eqns. 2.30 to 2.32. For the series of training outputs $\boldsymbol{Y}$ and the corresponding inputs $\boldsymbol{X}$ of length N, the covariance can be shown in index notation by Eqn. 2.33 and Eqn. 2.34 shows

the matrix form of the equation. Thus, the covariance is dependent upon the noise of the samples. In this work the kernel function used is a squared exponential function for which a different length scale is used for the different inputs of $\vec{x}$ as shown by Eqn. 2.35. For the problem of determining the propeller force coefficient in terms of J and β, the length of the feature vector length d_g is two. σ_f, l, and σ_n are free parameters that can either be solved for or specified. The $\boldsymbol{K}$ is an N by N matrix, thus specifying a GPR in the a customized CFD package requires specifying a matrix with size equal to the number of training points.

$$f(\vec{x}) \sim \mathcal{GP}(m(\vec{x}), k(\vec{x}, \vec{x}')) \tag{2.30}$$

$$m(\vec{x}) = E[f(\vec{x})] \tag{2.31}$$

$$k(\vec{x}, \vec{x}')) = E[f(\vec{x} - m(\vec{x}))(f(\vec{x}') - m(\vec{x}'))] \tag{2.32}$$

$$cov(y_q, y_p) = k(\vec{x_p}, \vec{x_q}) + \sigma_n^2 \delta_{pq} \tag{2.33}$$

$$cov(\boldsymbol{Y}) = \boldsymbol{K}(\boldsymbol{X}, \boldsymbol{X}) + \sigma_n^2 \boldsymbol{I} \tag{2.34}$$

$$k(\vec{x_p}, \vec{x_q}) = \sigma_f^2 \; exp\left[-\frac{1}{2}\sum_{m=1}^{d_g} \frac{(x_{p,m} - x_{q,m})^2}{l^2}\right] \tag{2.35}$$

2.5 A Simple 1-D Case: Open Water Propeller Thrust Coefficient

The open water relationship between K_T and J can be used to illustrate the capabilities of linear regression and GPR. *Gaggero et al.* (2019a) illustrated how a GPR could be used to construct a model for an open water propeller using BEM and a high fidelity model as a demonstration for how GPR could be used for propeller modeling. In the study presented here, the open water propeller curve determined by *Knight and Maki* (2019) for different grid densities for RANS CFD simulations of

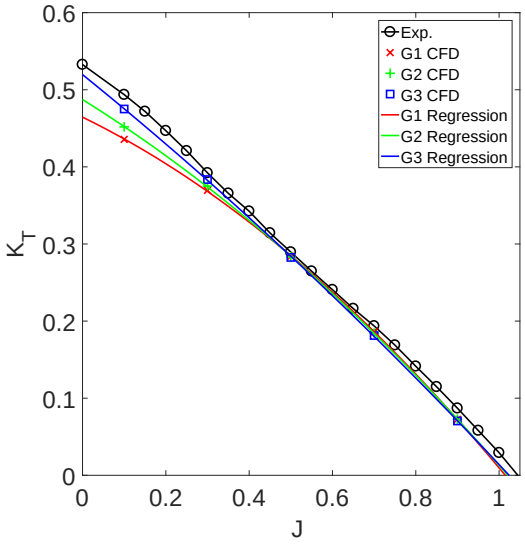

Figure 2.5: Linear regression prediction of open water propeller curve trained with CFD points of varying fidelity.

an open water propeller can be used and compared to experiments from *SIMMAN* (2008). The purpose of this illustrative example is to demonstrate the effect that the inherent error in the data points has upon a data-driven model.

In this simple case study, the model scale propeller for the KCS vessel is analyzed in open water at various values of J. Three RANS CFD grids are used by *Knight and Maki* (2019) at five different values of J are modeled using OpenFOAM version 2.4.x with the solver simpleFoam. The finest grid G3 has 7.2 million cells, the medium grid G2 has 1.3 million cells, and the coarsest grid has 0.3 million cells. To note, *Knight and Maki* (2019) show a Grid Convergence Index (GCI) using the method proposed by *Celik et al.* (2008), which indicates that the numerical uncertainty for the G3 grid is 3.27% on average. The Multiple Reference Frame (MRF) approach is used in a steady state simulation since the flow is axisymmetric. The $k-\omega$ SST turbulence model is used. The propeller is simulated at five different J values of 0.1, 0.3, 0.5, 0.7, and 0.9. A second-order linear regression model with features $[1, J, J^2]$ is used to model the K_T for each grid as shown by Fig. 2.5. In this figure, the experimental results are

shown along with the CFD calculations and the linear regression model prediction of K_T across the single parameter J. The fidelity of the CFD model increases with the size of the grid which scales with the computational cost of the model. Thus, the G3 grid is more accurate than the G1 grid, especially at the low values of J when the propeller operates off-design. Fig. 2.5 illustrates how the simple linear regression model captures the data that it is trained with well, however, for low-fidelity training data like that generated by the G1 grid, there is poor agreement with the experiment, especially at low values of J. On the other hand, the regression model trained with the G3 grid agrees well with the experiment, except for some deviation at low values of J, which could be corrected with even higher fidelity training data. Thus, in this simple one dimensional case, the linear regression model performs well at predicting the K_T as a function of J for the data that it is trained with.

Similarly, a GPR model is used to predict the K_T as a function of the input J. Fig. 2.6 shows the GPR prediction for each set of data, as well as the 95% prediction interval. The models trained with the G1 grid are at the top, the models trained with the G2 grid are in the middle, and the models trained with the G3 grid are on the bottom. The left hand column of images in this figure show the predictions if σ_n is assumed to be nearly zero, such that the training points based upon CFD are truth. When a σ_n of zero is used, the GPR prediction exactly predicts the correlating CFD prediction that is used for training and the result is very similar to that found with the linear regression model.

On the other hand, σ_n can be specified to a non-zero number correlating to the standard deviation of the error of each respective CFD model relative to the experimental results, such that σ_n can be determined as a function of the error between the CFD and the experimental prediction ϵ_d. For the purposes of this study, the experimental values are taken as truth, and thus the ϵ_d for each point is simply the difference between the experimental value and the CFD prediction. Eqn. 2.36 shows

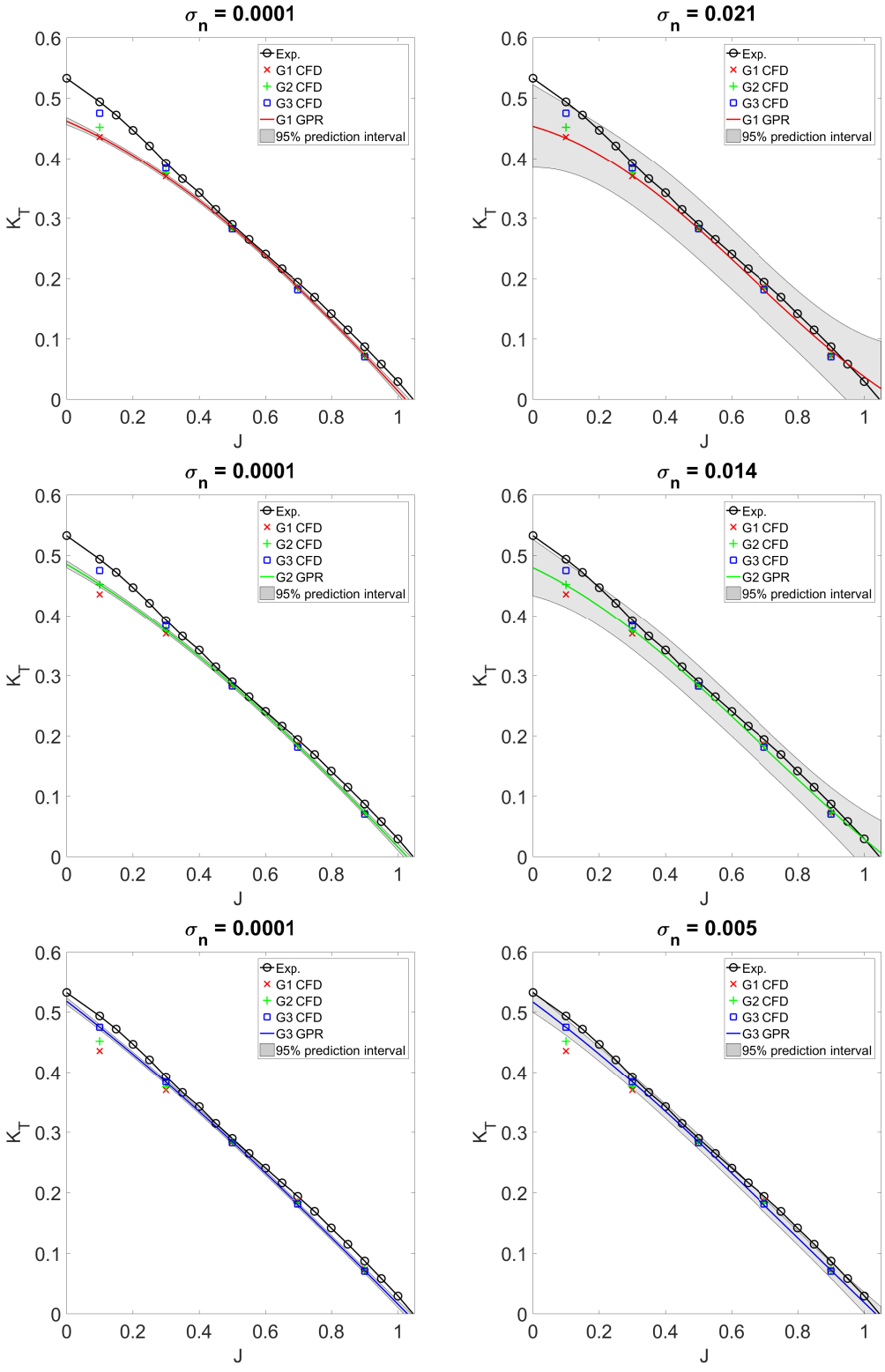

Figure 2.6: GPR prediction of open water propeller curve trained with CFD points of varying fidelity. The fidelity of the training model increases in the figures from from top to bottom. The left hand figures show results for a low σ_n, while the right shows results for a calculated σ_n.

how the sample standard deviation of the error can be determined as a function of the number of samples N, the discretization error ϵ_d, and the mean of the error μ_ϵ across all samples. For a sample size as small as is examined here there is some bias that would be removed for a larger sample set. For this reason, the prediction interval provided by the GPR does not encapsulate the experimental calculation at the lowest J. When the σ_n is calculated from the error of the training data the size of the prediction interval increases, thus, the model trained with the G3 grid has a much smaller prediction interval than the model trained with G1 grid. Additionally, the prediction interval is smaller near the training points and increases further away from the training points. This prediction interval is useful since it can indicate where points should be sampled to improve the fidelity of the model, and when a calculated σ_n is used it can indicate if higher fidelity simulations are needed. In the case examined here, for the G1 and G2 grids the uncertainty interval grows at the low values and high values of J.

$$\sigma_n = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(\epsilon_{d,i} - \mu_\epsilon)^2} \tag{2.36}$$

This simple case also illustrates the viability of multi-fidelity data-driven modeling, whereby a low fidelity tool can be used in the regime in which it is accurate, while a higher fidelity tool can be used when the propeller operates off design. For example, when the propeller operates on-design the coarse grid could be used, but at low values of J the propeller operates off-design and a higher fidelity grid like the G3 grid should be used. Commensurately, this could apply in the larger context where tools like BEMT can be used on-design, but viscous flow solvers like RANS CFD should be used for off design points. For heavily separated flows DES or LES could also be used. To illustrate this concept Fig. 2.7 shows a linear regression model and a GPR model prediction for which the G3 grid is used to train the points for J

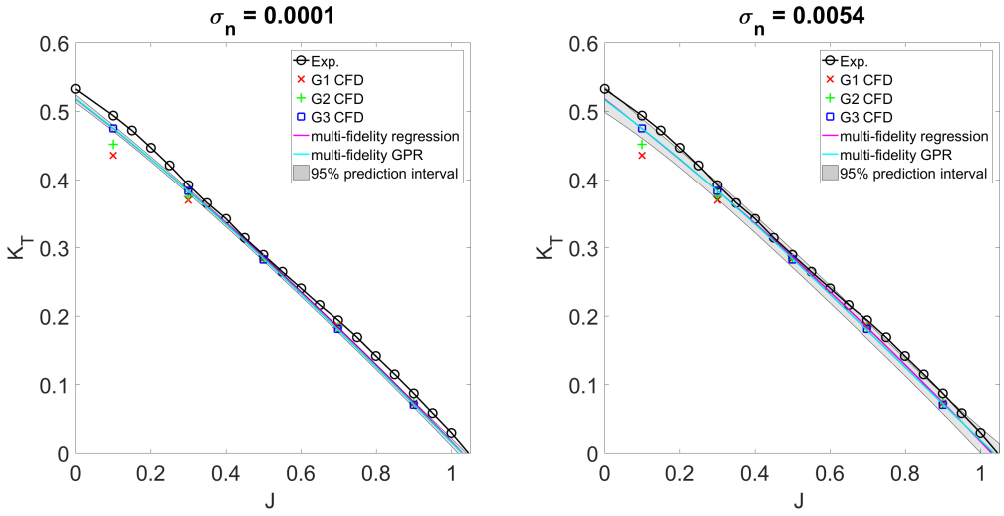

Figure 2.7: Linear regression and GPR prediction of open water propeller curve trained with CFD points trained with multi-fidelity. The left hand figure show results for a low σ_n, while the right hand figure shows results for a calculated σ_n.

values of 0.1 and 0.3, while the G1 grid is used for the other three values of J. The practitioner of the method can use engineering judgment to determine what tool is best used at a given condition. In this example case, either linear regression or GPR could be used to get results very similar to a method trained with only the G3 grid, but at a significantly reduced cost, since only 40% of the finest grid simulations are required. This same approach could be applied to more complicated problems with higher dimensions, like determining the forces on both the propeller and rudder for a maneuvering simulation. Lower fidelity methods could be used when attached flow is expected, and higher fidelity methods can be used to train data-driven models when the flow is more complex.

This simple one dimensional test case of determining K_T as a function of J is meant to illustrate how a data-driven problem with a simple scope is dependent upon both the fidelity of the data-driven method itself, but also upon the data that is used to train and evaluate the model. This study illustrates that both linear regression and Gaussian process regression are suitable methods for constructing a simple one

dimensional data-driven model based on CFD results. A benefit of the Gaussian process regression model is that it provides a confidence interval which could be useful for indicating where else should be sampled in the parameter space, which could be useful for a more complex function. On the other hand, the implementation of linear regression is simpler and scales with the number of features used in the model. The implementation of Gaussian process regression in a custom CFD package is somewhat more complicated and scales with the number of training points. These principles are important to consider when constructing data-driven models based upon data that is challenging to model accurately, like the flow over the propeller and rudder for a maneuvering vessel.

CHAPTER III

Propeller and Rudder Data-Driven Model Development for the KRISO Container Ship (KCS)

The KRISO Container Ship (KCS) is a single screw container ship with a semi-balanced horn rudder. Model scale experiments have been performed for the *SIMMAN* (2020) workshop and these are used to evaluate the performance of the CFD method. Specifically, this work compares to Case 3.3 of the workshop for which the particulars of the geometry are shown in Table 3.1. This chapter shows how a data-driven model can be developed using select RANS CFD simulations of the propeller and deflected rudder operating in the behind condition. Different variations of the data-driven propeller and rudder model are evaluated by comparing the propeller and rudder force to the experimental values during the maneuver using the experimental motions of the vessel as input to the model. In Chapter IV a select number of these data-driven models for the propeller and rudder forces are further evaluated by performing a turning circle maneuver with the KCS at model scale in both calm water and in waves. Fig. 3.1 shows the vessel from two different perspectives along with the direction that the propeller rotates and the rudder deflection angle.

This vessel presents challenges for analysis using CFD methods with a discretized

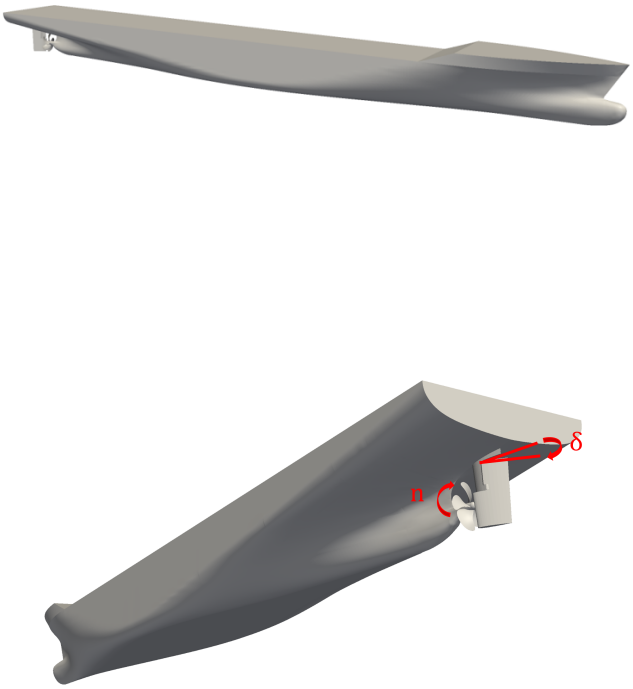

Figure 3.1: KCS geometry. Top: Vessel viewed front-starboard side. Bottom: Vessel viewed from underneath with propeller rotation direction and rudder angles shown.

Table 3.1: Parameters for KCS geometry and case setup.

Parameter	Value
Length Between Perpendiculars (L_{PP})	3.057 m
Model Scale (λ_s)	1/75.24
Initial velocity u_o	0.860 m/s
Rudder angle δ	35 °
Distance of center of gravity to rudder stock d_r	1.482 m
Displacement	0.122 m^3
Draft	0.144 m
Number of propeller blades	5
Propeller diameter D	0.105 m
Distance of center of gravity to propeller d	1.428 m

propeller and rudder. Like many vessels, the rudder is in close proximity to the propeller therefore fitting a rotating sliding mesh region for both the propeller and the rudder is infeasible. Additionally, a sliding mesh approach does not work because the stationary part of the rudder and the moving part of the rudder cannot be separated with this approach. To address this, the geometry must be either simplified as performed in *Piro et al.* (2020) to enable the use of a sliding mesh or a different approach such as an overset grid or immersed boundary method could be used, but these add computational complexity. This work removes this challenge by using a data-driven propeller and rudder model, which removes the complexity of multiple moving regions in the CFD simulation.

The model development is discussed in terms of the CFD technique used as well as how the data-driven model is constructed and validated. Double-body RANS CFD grids with the propeller operating in the behind condition with the deflected rudder are used to train the data-driven model. This data-driven model is applied to VOF simulations of the vessel performing a turning circle maneuver both in calm water and in waves.

3.1 CFD Model Development

OpenFOAM version 2.4.x is used for all CFD in this work. The background grids are generated using the OpenFOAM utility blockMesh and the vessel geometry is incorporated using snappyHexMesh. Two separate CFD models are created; one is for training the data-driven model and the other is for performing maneuvering simulations using the data-driven model. The CFD model used for the training of the data-driven model uses a double-body approximation for which the water-plane is modeled as a symmetry plane. For the maneuvering CFD simulations the VOF approach is used to capture the interface between the water and air (described in Chapter IV).

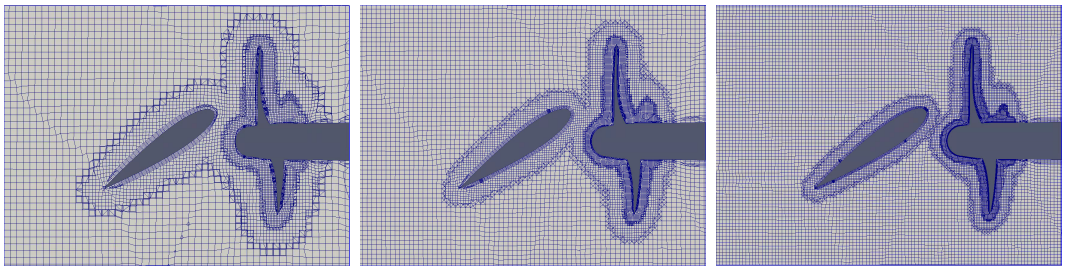

Figure 3.2: Slice of KCS double-body grid at the propeller plane (0.089 m) below the water-plane with Z normal. Left: G1 grid. Middle: G2 grid. Right: G3 grid

3.1.1 Double-Body CFD of Discretized Propeller and Rudder

The double-body RANS CFD of the hull, discretized propeller and rudder is run using the OpenFOAM solver pimpleDyMFoam. The horned rudder is deflected to an angle δ of 35°. Three grids are examined. The G1 grid has 2.0 million cells, the G2 grid has 5.2 million cells, and the G3 grid has 14.1 million cells. Fig. 3.2 shows a slice through the domain with Z normal at the propeller rotation axis for each of the grids. The propeller rotates at n=10.4 rps and the duration of each simulation is 5.77 s which correlates to 60 propeller revolutions. During the simulation, the propeller is rotated at no more than one degree per time-step using a rotating sliding mesh.

The CFD domain is 4 L_{PP} long, 3 L_{PP} wide, and 1 L_{PP} deep. The vessel is located at the top of the domain in the longitudinal and lateral center of the domain. The top of the domain is a plane of symmetry. The bottom of the domain, the side at maximum value of positive Y, and the side at the maximum positive X are velocity inlets. The sides of the domain at the maximum value of negative Y and negative X are outlets. The hull, propeller, and rudder are non-slip walls modeled with wall functions. The bow is pointed in positive X; positive Y is towards port; positive Z is vertical upwards. Figure 3.3 shows the CFD domain on the left as well as the surface grid of the propeller and rudder on the right. In the image the vessel is shown in red.

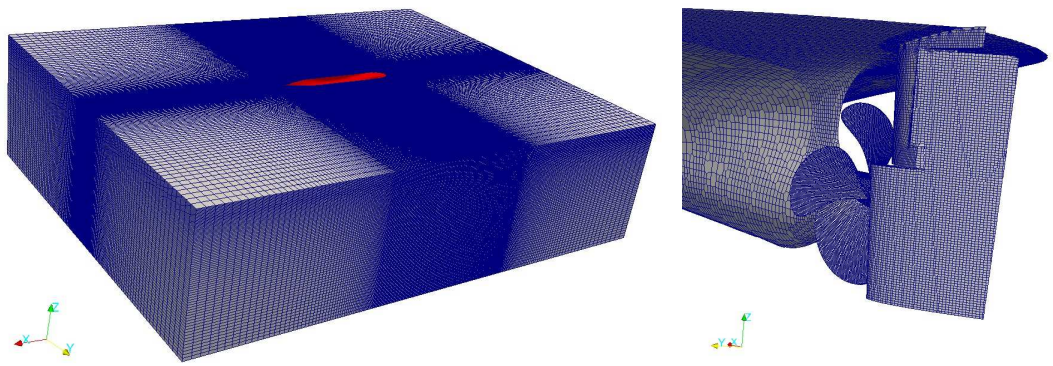

Figure 3.3: KCS Double Body G2 mesh. Left: Domain for double-body simulations of KCS. Right: Surface grid on KCS propeller and rudder.

In the region around the hull, extending +/− 1.86 m (0.6 L_{PP}) fore and aft from the vessel center of gravity, 0.32 m laterally in each direction, and from the waterline to 0.2 m below the waterline the cells of the grid are nearly isotropic, such that the cell discretization length in the X, Y, and Z directions are within 2% of the characteristic length of each cell Δ_g. Δ_g is the average of the cell length in X, Y, and Z directions. Outside of the region immediately surrounding the vessel, stretching is applied to the cells to reduce grid count as illustrated in Fig. 3.3. A refinement region is applied around the propeller and rudder to better capture the complex flow and three levels of refinement are applied using snappyHexMesh. Each level of refinement halves the Δ_g. Different levels of refinement are applied to each surface. One level of refinement is applied to the hull and in regions with large curvature a second level of refinement is applied, four levels of refinement are applied to the rudder, and five levels of refinement are applied to the propeller. Five prism layers with a thickness of 0.3 of the local Δ_g are applied to each surface. Table 3.2 shows the mesh size on each grid for the different regions and surfaces.

Table 3.2: KCS double-body CFD grid details.

Grid	G1	G2	G3
Isotropic region $\Delta_g(m)$	2.86E-02	2.02E-02	1.43E-02
Refinement zone Δ_g (m)	3.58E-03	2.53E-03	1.79E-03
Hull Δ_g (m)	1.43E-02	1.01E-02	7.15E-03
Rudder Δ_g (m)	1.79E-03	1.26E-03	8.94E-04
Propeller Δ_g (m)	8.94E-04	6.32E-04	4.47E-04

3.2 Data-Driven Model Development

The double-body CFD simulations with a discretized propeller and rudder are used to train a series of data-driven models. Data-driven models are constructed to determine the K_T, K_S, K_Q, C_{Fx}, and C_{Fy}. As noted in Chapter II, two methods for generating data-driven models are examined: linear regression and Gaussian process regression. In this chapter these two methods are compared. Beyond the effect of what method is used to train a data-driven model, the effect of the fidelity of the data used to train the data-driven model is studied in detail. The fidelity of the data is measured by comparing different CFD grid densities and the effect on the underlying discretization error. Additionally, two different turbulence models are used in the double-body CFD, and the differences between the prediction of force on the propeller and rudder is discussed.

The effect of the number of sample points used to train the models are also considered. The J-β space is sampled with bounds of J between 0.2 and 0.9, and β between 0° and 45°. This bounding box in two-dimensional space, which encompasses the expected J-β bounds while the vessel performs a turning circle maneuver, is sampled using Latin Hypercube Sampling (LHS). Three distinct LHS sets are generated to train and evaluate different data-driven models. One set has four sample points, one set has eight sample points, and the last set has sixteen sample points. Predominantly, the data split between training and validation is such that the four point data set is used for validation and the eight point and sixteen point sets are used to train

models, either individually or in combination. This set of data is shown in Fig. 3.4 in the $J - \beta$ space and the experimental results (*SIMMAN* (2020)) are shown in conjunction with the sample points in Fig. 3.5. The specific values are also shown in Tables A.1-A.3. Later, LHS points one through four are used primarily as validation data for the data-driven models, while LHS points five through twenty-eight are used for training the data-driven model. By using the four point set of data for validation, it makes it so that the data split between the training and validation data is two thirds of the data for training and one third of the total data for validation when the eight point set of data is used to train the model. On the other hand, the data split between training and validation data is: 80% training and 20% validation when the sixteen point set of data is used to train the model. By always using the same set of data for validation it creates a constant comparison point for when different sets of data are used.

In the double-body CFD analysis used to train the model, the propeller revolution rate is held constant at n 10.4 rps and the velocity is specified at the inlet, such that the vessel operates at the desired drift angle. Since only forward velocity and sway velocity are considered, β, β_P, and β_R are all equal. This simplifies the training of the method, since only β is necessary to train the models, but the yaw rate is accounted for in the CFD maneuvering simulations (for the calculation of β_P and β_R) in Chapter IV.

The goal of the data-driven model is to determine the multi-degree-of-freedom force acting on both the propeller and rudder based upon the value of J and β calculated in radians per second. To train the model, the double body CFD cases are calculated and the forces on the propeller and rudder are averaged over the second half of each simulation. The y^+ for LHS sample point 2 is in Table A.4. The flow field is complex since both the propeller and rudder operate in the wake of the hull, and the rudder operates in the wake of the propeller as shown by Figure 3.6. This

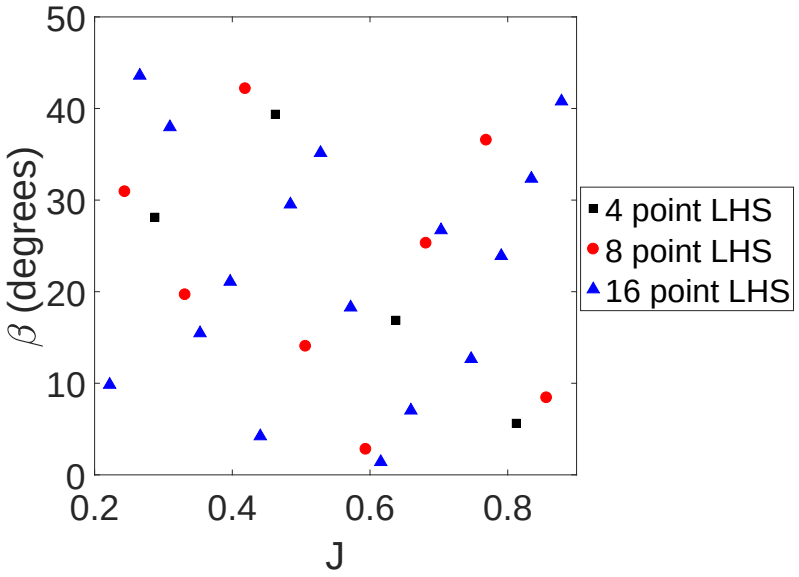

Figure 3.4: LHS points for training the KCS data-driven model.

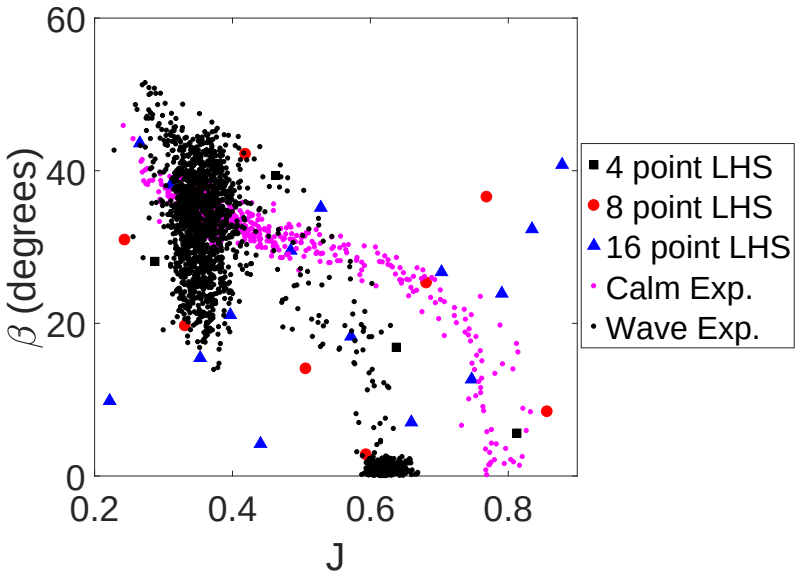

Figure 3.5: LHS points for training the KCS data-driven model with experimental data (*SIMMAN* (2020)). The experimental data points show β_P.

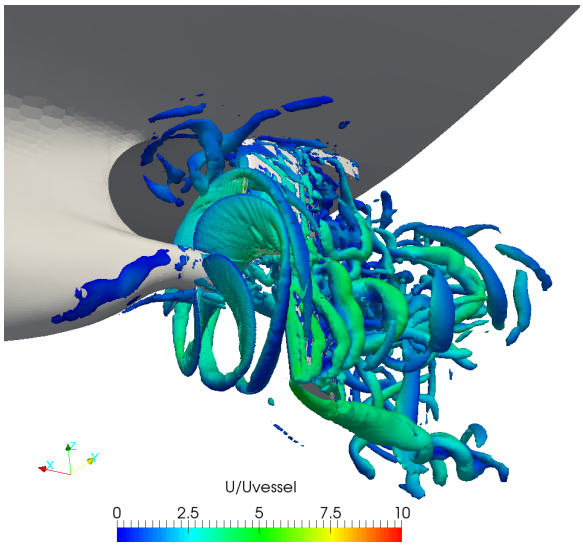

Figure 3.6: Q criterion of 2500 contour for G3 LHS point 11 illustrating the propeller-rudder interaction.

image illustrates the complex flow field around the propeller and rudder which shows the importance of modeling this phenomenon with high fidelity CFD.

The data-driven models trained with CFD simulations contain several types of error which determine the total error in the data-driven model ϵ_m. These include: discretization error ϵ_d, turbulence modeling error ϵ_T, and data modeling error ϵ_r. The discretization error is an error based in the CFD mesh discretization and can be categorized by comparing the forces predicted with the different grid densities. The turbulence modeling error can be categorized by comparing the forces predicted for different turbulence modeling techniques, but it is also influenced by the wall functions used and discretization error. For this case we examine the use of both the Spalart-Allmaras (SA) turbulence model as well as the $k-\omega$ SST turbulence model. The data-driven modeling error can be determined by calculating how well the model predicts the data that it is trained on as well as how well it predicts the validation data.

The total error of the model can be determined by the difference between the truth value $\hat{y}$ and the model prediction y as shown by Eqn. 3.7. The truth is not

known. The closest that the truth can be predicted by is either experimental results or the finest CFD results, both of which still contain error. In this study, the results using the model are compared to both experimental results and when experimental results are unavailable fine CFD results are used.

$$\epsilon_m = \hat{y} - y \tag{3.1}$$

3.2.1 Discretization Error

To compare the mesh discretization error the calculated forces are compared between the different grids (G1, G2, G3) for different samples. Tables 3.3, A.5, A.6, and 3.4 show the values and discretization percent error relative to the finest grid for the K_T, $10K_Q$, C_{Fx}, and C_{Fx} respectively. These tables show the results for LHS points one through twelve. Oscillatory convergence is present for many of the cases so quantifying the discretization error with Richardson extrapolation is not possible. *Eca and Hoekstra* (2009) indicate that if a CFD study does not have monotonic convergence, then the discretization error can be estimated by three times the maximum difference of that quantity across all three grids. In this study the discretization error is characterized by calculating the relative error of the force coefficients from G1 and G2 grids with respect to the G3 results as shown by Eqns. 3.2-3.3 similar to *Celik et al.* (2008). ϵ_{13} denotes the error for the G1 grid relative to the G3 grid prediction, in terms of the G1 grid prediction ϕ_1 and the G3 grid prediction ϕ_3; similarly, ϵ_{23} denotes the error for the G2 grid prediction relative to the G3 grid prediction in terms of the G2 grid prediction ϕ_2 and ϕ_3. The average of the absolute value of the ϵ_{13} and ϵ_{23} across a data-set is used as the measure of discretization error for that data-set and is denoted $\overline{|\epsilon_{i3}|}$. Thus for a data-set with N samples calculated with grid i (for i equal to 1 or 2), the average of the absolute value of the relative error (with respect to G3) across all N samples is denoted $\overline{|\epsilon_{i3}|}$.

Table 3.3: Discretization error between different grids for K_T.

LHS point	G1 (K_T)	G2 (K_T)	G3 (K_T)	ϵ_{23} (%)	ϵ_{13} (%)
1	0.321	0.330	0.327	-0.933	1.697
2	0.384	0.407	0.404	-0.905	4.852
3	0.281	0.272	0.266	-2.069	-5.508
4	0.252	0.237	0.242	1.924	-4.267
5	0.253	0.247	0.235	-5.019	-7.533
6	0.297	0.309	0.316	2.129	5.865
7	0.408	0.423	0.420	-0.531	2.929
8	0.334	0.340	0.339	-0.308	1.604
9	0.198	0.194	0.183	-5.887	-8.068
10	0.330	0.320	0.322	0.474	-2.670
11	0.394	0.394	0.390	-0.968	-0.967
12	0.238	0.214	0.206	-3.855	-15.205

$$\epsilon_{13} = \frac{\phi_3 - \phi_1}{\phi_3} \tag{3.2}$$

$$\epsilon_{23} = \frac{\phi_3 - \phi_2}{\phi_3} \tag{3.3}$$

The average magnitude of the K_T percent error between the G3 and G2 grids is 2.08% while the average magnitude of the percent error between the G3 and G1 grids is 5.10%. The average percent error in $10K_Q$ is 2.20% between G3 and G2 but increases to 4.49% between G3 and G1. If only the LHS points one through four are analyzed the average magnitude of the K_T percent error between the G3 and G2 grids is 1.46% while the average magnitude of the percent error between the G3 and G1 grids is 4.08%. Similarly for $10K_Q$ the percent error between the G3 and G2 grids for LHS points one through four is 1.55%; while the error increases to 3.60% when G3 and G1 are compared for LHS points one through four. Fig. 3.7 illustrates the average discretization error for K_T and K_Q for each grid.

There is more discretization error for the forces on the rudder compared to the

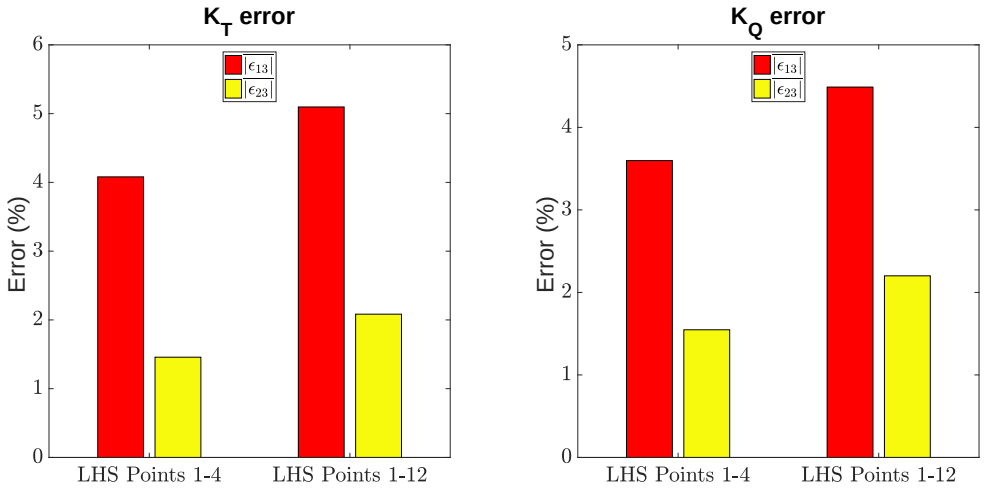

Figure 3.7: Average absolute value of the discretization error ($|\overline{\epsilon_{i3}}|$) for K_T (left) and K_Q (right) for G1, G2, and G3. The average error is presented for the validation set (LHS points one through four) as well as for LHS points one through twelve.

forces on the propeller. The average percent error in C_{Fx} is 3.87% between G3 and G2; this increases to 8.35% between G3 and G1. Similarly, the average percent error in C_{Fy} is 8.39% between G3 and G2; this increases to 16.72% between G3 and G1. The maximum error on the rudder forces occurs for LHS point 12, which correlates to the highest value of J and a relatively low value of β. For this point the flow is challenging to model since the rudder operates at a high angle of attack when β is low and the rudder is deflected to 35°. Thus, it is sensible that the discretization error is high when the flow is highly separated. If only the LHS points one through four are analyzed the average magnitude of the C_{Fx} percent error between the G3 and G2 grids is 3.94% while the average magnitude of the percent error between the G3 and G1 grids is 8.12%. Similarly for C_{Fy} the percent error between the G3 and G2 grids for LHS points one through four is 9.96%; while the error increases to 18.11% when G3 and G1 are compared for LHS points one through four. Fig. 3.8 illustrates the average discretization error for C_{Fx} and C_{Fy} for each grid.

Table 3.4: Discretization error between different grids for C_{Fy}.

LHS point	G1 (C_{Fy})	G2 (C_{Fy})	G3 (C_{Fy})	ϵ_{23} (%)	ϵ_{13} (%)
1	0.244	0.265	0.274	3.176	10.858
2	0.245	0.261	0.278	6.282	12.015
3	0.330	0.383	0.453	15.483	27.162
4	0.387	0.424	0.498	14.923	22.406
5	0.334	0.338	0.380	11.023	12.157
6	0.368	0.395	0.446	11.328	17.362
7	0.241	0.245	0.250	2.278	3.623
8	0.222	0.241	0.255	5.638	13.089
9	0.174	0.219	0.223	2.034	22.043
10	0.306	0.367	0.403	9.030	24.100
11	0.298	0.316	0.313	-0.768	4.876
12	0.362	0.426	0.525	18.735	31.007

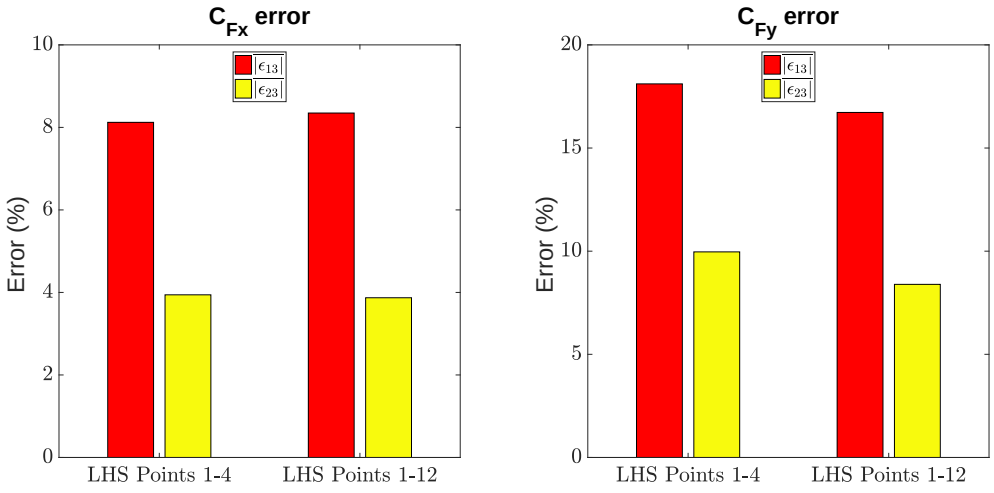

Figure 3.8: Average absolute value of the discretization error ($|\overline{\epsilon_{i3}}|$) for C_{Fx} (left) and C_{Fy} (right) for G1, G2, and G3. The average error is presented for the validation set (LHS points one through four) as well as for LHS points one through twelve.

3.2.2 Turbulence Modeling Error

The turbulence modeling error is evaluated by comparing the results using the Spalart Allmaras versus the $k-\omega$ SST turbulence models for a given LHS point for a given mesh discretization. Thus, the relative error for a quantity ϕ is determined as a function of the prediction with the Spalart Allmaras turbulence model ϕ_{SA} and the prediction with the $k-\omega$ SST turbulence model $\phi_{k-\omega\ SST}$ can be denoted ϵ_T and is defined by Eqn. 3.4. The average of the absolute value of the ϵ_T for a data-set for a given grid is used as the measure of turbulence modeling error across the data-set and is denoted $\overline{|\epsilon_T|}$. Thus for a data-set with N samples calculated for a given grid, the average of the absolute value of the ϵ_T across all N samples is denoted $\overline{|\epsilon_T|}$.

$$\epsilon_T = \frac{\phi_{SA} - \phi_{k-\omega\ SST}}{\phi_{SA}} \tag{3.4}$$

The four point LHS set is run with both turbulence models for all three training grids. The calculated value for each case and the ϵ_T in K_T, $10K_Q$, C_{Fy}, and C_{Fy} are shown in Tables 3.5, A.7, A.8, and 3.6 respectively. These tables depict the values of the force coefficients predicted with each turbulence model for each respective grid and LHS point as well as the percent error of the difference between the two turbulence models with respect to the Spalart Allmaras turbulence model results.

The best agreement between the turbulence models for the G2 and G3 grids for the calculation of K_T and K_Q occurs with LHS point one while the worst agreement occurs for LHS point three. In the flow conditions for LHS point one the J is relatively low but the β is high which leads to a low angle of attack acting on the rudder. The rudder is less heavily loaded for LHS point one relative to LHS points three and four which leads to less effect of the rudder induced velocity on the propeller. Another means of evaluating the performance of the propeller is to determine the propeller efficiency η_p as shown by Eqn. 3.5. The propeller is most efficient near its design

Table 3.5: Turbulence modeling error between Spalart Allmaras and $k-\omega$ SST for K_T.

Grid	LHS point	Spalart Allmaras (K_T)	$k-\omega$ SST (K_T)	ϵ_T (%)
G1	1	0.321	0.328	-2.186
G2	1	0.330	0.330	-0.133
G3	1	0.327	0.324	0.812
G1	2	0.384	0.412	-7.308
G2	2	0.407	0.423	-3.829
G3	2	0.404	0.421	-4.303
G1	3	0.281	0.319	-13.597
G2	3	0.272	0.317	-16.518
G3	3	0.266	0.317	-18.801
G1	4	0.252	0.248	1.503
G2	4	0.237	0.254	-7.302
G3	4	0.242	0.252	-4.390

condition which occurs at an open water advance coefficient of approximately 0.8 (*SIMMAN* (2020)). At low values of J the propeller is more heavily loaded and becomes less efficient. For LHS point 3, the $k-\omega$ SST turbulence model predicts more thrust but commensurately less torque than the Spalart Allmaras turbulence model for all grids examined. This leads to the propeller operating more efficiently for all grids with the $k-\omega$ SST turbulence model. If the average of of the magnitude of η_p is calculated for all grids, the Spalart Allmaras turbulence model predicts that the propeller is 3.76% less efficient for LHS point 3 than the $k-\omega$ SST turbulence model. A similar phenomenon occurs for LHS point 2 which is the lowest J value examined and is also the least efficient data point in this set as calculated with both turbulence models. For LHS points one and four, the G1 and G2 calculation have a higher efficiency with the $k-\omega$ SST turbulence model but G3 is more efficient with the Spalart Allmaras turbulence model.

$$\eta_p = \frac{JK_T}{2\pi K_Q} \tag{3.5}$$

Table 3.6: Turbulence modeling error between Spalart Allmaras and $k-\omega$ SST for C_{Fy}.

Grid	LHS point	Spalart Allmaras (C_{Fy})	$k-\omega$ SST (C_{Fy})	ϵ_T (%)
G1	1	0.244	0.187	23.618
G2	1	0.265	0.195	26.475
G3	1	0.274	0.192	30.025
G1	2	0.245	0.254	-3.721
G2	2	0.261	0.287	-9.897
G3	2	0.278	0.288	-3.330
G1	3	0.330	0.311	5.679
G2	3	0.383	0.310	19.047
G3	3	0.453	0.374	17.421
G1	4	0.387	0.445	-15.155
G2	4	0.424	0.373	12.163
G3	4	0.498	0.374	24.982

The turbulence modeling error is a larger source of error on average than the difference in calculated values between different grids for the calculation of K_T and $10K_Q$. For LHS points one through four the average of the absolute value of the percent error for K_T calculated with the Spalart Allmaras turbulence model between G3 and G2 is 1.46%, while the percent error between G3 and G1 is 4.08%. The turbulence modeling error between the two turbulence models is 6.15% for G1, 6.94% for G2, and 7.08% for G3. Similarly for the same LHS points the average of the absolute value of the percent error for $10K_Q$ calculated with the Spalart Allmaras turbulence model between G3 and G2 was 1.55%, while the percent error between G3 and G1 was 3.60% on average. The turbulence modeling error between the two turbulence models is 3.92% for G1, 4.70% for G2, and 7.05% for G3.

The prediction for the rudder force coefficients differ significantly between the two turbulence models examined. Depending upon the turbulence model used the separation point for flow over a foil can vary and when separation occurs the forces can change radically for small differences in angle of attack. Furthermore, the two turbulence models used in this work assume a fully turbulent boundary layer, but

due to the small scale of the vessel, a transitional turbulence model may be more accurate for determining the propeller and rudder forces; however, using a transitional turbulence model would require finer wall-resolved grids. For all grids and LHS points, except the G1 LHS point four case, the Spalart Allmaras turbulence model predicts a larger magnitude of C_{Fx} compared to the $k-\omega$ SST turbulence model. Table 3.6 shows that for LHS points one and three the Spalart Allmaras turbulence model predicts a higher C_{Fy} than the $k-\omega$ SST turbulence model, whereas for LHS point 2 the opposite occurs. Mixed results for LHS point four are seen for the calculation of C_{Fy} depending upon which grid is used. The magnitude of the percent error is largest for the calculation of LHS point one with all three grids for both C_{Fx} and C_{Fy}. Conversely, the magnitude of the percent error for LHS point two is the lowest for this data set. LHS point two correlates to the lowest rudder force coefficients for the Spalart Allmaras turbulence model, whereas lowest rudder force coefficients for the $k-\omega$ SST turbulence model occur at LHS point one. The highest rudder forces occur with LHS point four, which follows intuition since this is the highest J value and lowest β value such that the rudder operates a high angle of attack; however the two turbulence models are not in very good agreement at this data point. With the Spalart Allmaras turbulence model as the grid is refined the C_{Fx} and C_{Fy} increase in magnitude, whereas the G1 grid predicts the highest force when the $k-\omega$ SST turbulence model is used at this LHS point. Another important item to note is that the Spalart Allmaras turbulence model tends to have more variation in the prediction of rudder force across different grids for a given LHS point compared to the $k-\omega$ SST turbulence model.

3.2.3 Data-Driven Model for KCS Propeller and Rudder

Two variations of data-driven models are generated for the KCS propeller and rudder. The first approach generates a linear regression model and the second generates

a model using Gaussian process regression. The linear regression model is generated using the MATLAB function `fitlm`. The Gaussian process regression model is generated using the MATLAB function `fitgrp`. Data-driven models using these techniques are generated for the propeller thrust, propeller side force, propeller torque, rudder side force, and rudder resistance force. Two separate sets of data-driven models are generated for the rudder: the first accounts for both the movable and stationary parts of the rudder while the second only accounts for the movable part of the rudder. When the propeller and rudder model are implemented for CFD based maneuvering simulations, the rudder model that accounts for the force on the whole rudder (both movable and stationary parts) is used. On the other hand, a separate model for just the movable part of the rudder is developed since the experimental results (*SIMMAN* (2020)) report the rudder normal force FN on the movable part of the rudder, but not on the stationary part.

The diameter of the propeller is 0.105 m and the propeller revolution rate is 10.4 rps. In Chapter IV, when the propeller and rudder model are implemented in the CFD simulation for the maneuvering of the vessel, the full lateral area of the rudder S_R of 0.0096 m^2 is used. This S_R accounts for both the stationary and movable parts of the semi-horned rudder. This S_R is used for evaluating the error of the linear regression model with respect to the training data and subsequently the comparison between different linear regression model response surfaces.

The data modeling error with linear regression is examined in the context of the underlying discretization error and turbulence modeling error. Subsequently, the effect of using different sets of training data for the linear regression model is examined. The response surfaces generated with linear regression versus Gaussian process regression are compared.

For the comparison of the rudder normal force predicted using data-driven modeling predictions, the S_R used is that of only the movable part of the rudder which is

$0.008\ \mathrm{m}^2$ and the rudder force of only the movable part of the rudder is accounted for. An analysis is also performed for which the linear regression model and the Gaussian process regression model are compared.

3.2.4 Data Modeling Error With Regression

A data-driven model can be evaluated based on how well it predicts validation data. The four point LHS set is used to validate the models. The eight point and sixteen point sets of data are used to train the models. Thus the ratio of training to validation data when the eight point set is used is 2:1, whereas when the sixteen point set of data is used to train the model, the ratio is 4:1. A set of tables demonstrate the error of the ϵ_r, which is the regression model prediction of specific force parameters relative to the CFD prediction at each respective sample. The model is evaluated based on its accuracy to the CFD grid density that it was trained with; for example, the regression models that are trained with the G1 grid are compared to the G1 grid CFD predictions and the regression models that are trained with the G2 grid are compared to the G2 grid CFD predictions. Thus, for a model prediction ϕ_m and a CFD based calculation which is treated as a validation sample ϕ_V, the ϵ_r can be calculated by Eqn. 3.6. The average of the absolute value of the ϵ_r for a data-set is used as the measure of regression modeling error for a given validation data-set and is denoted $\overline{|\epsilon_r|}$. Thus for a data-set with N validation samples, the average of the absolute value of the ϵ_r across all N samples is denoted $\overline{|\epsilon_r|}$. The regression models use different sets of training data including the eight point LHS set and the sixteen point LHS sets for the G1 and G2 double-body training grids, as well as the eight point LHS set for the G3 double-body training grid. In this study of the modeling error, all cases use the Spalart-Allmaras turbulence model for training and validating the models. All of these regression models are constructed using the feature vector shown by Eqn. 2.25.

Table 3.7: Modeling error for K_T.

LHS Point	8 LHS G1 ϵ_r (%)	16 LHS G1 ϵ_r (%)	8 LHS G2 ϵ_r (%)	16 LHS G2 ϵ_r (%)	8 LHS G3 ϵ_r (%)
1	1.621	5.351	2.652	-0.021	2.621
2	-4.191	-4.526	0.470	1.111	0.434
3	-1.682	-0.495	-0.332	-0.411	-0.040
4	1.715	3.859	1.596	6.394	4.899

$$\epsilon_r = \frac{\phi_V - \phi_m}{\phi_V} \tag{3.6}$$

Table 3.7 shows the percent error of the prediction of K_T using the regression based prediction relative to the CFD calculation at each validation data point. The average value of the absolute error for each model can be calculated. All of the models predict the K_T within 4% on average; the worst performing model on average is the sixteen point LHS set trained with the G1 grid. Fig. 3.4 shows that some of the samples are quite close between the four point and eight point LHS sets, which may contribute to the improved prediction with the eight point set compared to the sixteen point set. The model trained with the eight point LHS set on the G2 grid is the most accurate with 1.26% error on average relative to the G2 grid CFD prediction.

Similarly, Table 3.8 shows the percent error of the prediction of K_Q using the regression based prediction relative to the CFD calculation at each LHS validation data point. The average value of the absolute error for each model shows that all of the models predict the K_Q within 3%; the worst performing model on average is the sixteen point LHS set trained with the G1 grid. The model trained with the eight point LHS set on the G2 grid is the most accurate with 0.74% error on average relative to the G2 grid CFD prediction.

The error magnitude is larger for the rudder force coefficient predictions compared to the propeller force coefficient predictions. Table 3.9 and Table 3.10 show the percent error of the prediction of C_{Fx} and C_{Fy} respectively using the regression based

Table 3.8: Modeling error for $10K_Q$.

LHS Point	8 LHS G1 ϵ_r (%)	16 LHS G1 ϵ_r (%)	8 LHS G2 ϵ_r (%)	16 LHS G2 ϵ_r (%)	8 LHS G3 ϵ_r (%)
1	1.264	4.183	1.554	0.002	1.925
2	-3.542	-3.814	0.296	0.891	0.225
3	-1.184	-0.390	-0.081	-0.108	-0.231
4	1.437	2.425	1.012	4.293	3.641

Table 3.9: Modeling error for C_{Fx}.

LHS Point	8 LHS G1 ϵ_r (%)	16 LHS G1 ϵ_r (%)	8 LHS G2 ϵ_r (%)	16 LHS G2 ϵ_r (%)	8 LHS G3 ϵ_r (%)
1	2.603	0.823	1.281	3.630	1.394
2	-3.782	-7.178	-2.820	-2.366	-1.056
3	-3.007	-2.526	-3.001	0.543	-0.709
4	-0.760	7.022	-1.697	-0.488	-5.131

prediction relative to the CFD calculation at each LHS validation data point. The average value of the absolute error for each model shows that all of the models predict the C_{Fx} within 5% and the C_{Fy} within 7%; the worst performing model on average is the sixteen point LHS set trained with the G1 grid. The model trained with the sixteen point LHS set on the G2 grid is the most accurate on average for the prediction of C_{Fx} with 1.75% error on average relative to the G2 grid CFD prediction. The model trained with the eight point LHS set on the G3 grid leads to the most accurate prediction relative to the CFD model it was trained with for the prediction of C_{Fy}, with less than 3% error. The model trained with the eight points on the G2 grid also leads to a reasonably good prediction of C_{Fx} and C_{Fy} with 2.20% and 3.29% error respectively.

Ultimately, the regression modeling error ϵ_r is small relative to the other sources of error including the discretization error and turbulence modeling error. As is shown in Tables 3.7-3.10, there is some modeling error induced by using the regression approach as compared to directly modeling it with CFD; however, the CFD has uncertainty as

Table 3.10: Modeling error for C_{Fy}.

LHS Point	8 LHS G1 ϵ_r (%)	16 LHS G1 ϵ_r (%)	8 LHS G2 ϵ_r (%)	16 LHS G2 ϵ_r (%)	8 LHS G3 ϵ_r (%)
1	1.494	1.364	3.599	14.385	0.457
2	-7.818	-12.055	-6.023	-1.353	-0.871
3	-7.423	-5.938	-0.998	2.141	2.556
4	1.609	7.713	-2.548	-7.238	-7.019

well. The data modeling error can be compared to the discretization and turbulence modeling errors as shown in Figs. 3.9-3.12. These figures compare the average of the absolute value of each source of error in the modeling of the LHS points one through four that were used to validate the model. In these figures the discretization error is shown as a percent error relative to the G3 grid for each respective force coefficient; this is shown by the red bars and the yellow bars for the $\overline{|\epsilon_{13}|}$ and $\overline{|\epsilon_{23}|}$ respectively. The turbulence modeling error ($\overline{|\epsilon_T|}$) is calculated based on the average of the absolute value of the difference in the calculation of each force coefficient for each respective grid between the two turbulence models examined; this is shown by the black bars. The data modeling error $\overline{|\epsilon_r|}$ is characterized by determining the average magnitude of the error for the validation points using the 8 point LHS set and 16 point LHS set for training each model respectively, these are denoted by `8 point LHS Model` $\overline{|\epsilon_r|}$ and `16 point LHS Model` $\overline{|\epsilon_r|}$ respectively in the figures. The $\overline{|\epsilon_r|}$ and $\overline{|\epsilon_T|}$ are shown for each respective grid (G1, G2, and G3), but the discretization error is shown separately since all cases contain discretization error. If the $\overline{|\epsilon_{33}|}$ were to be shown it would deceptively be 0% because the G3 model is treated as the 'truth' in this framework; but of course since there is oscillatory convergence there is indeed also uncertainty in the G3 grid CFD.

Fig. 3.9 shows the error terms for the prediction of K_T. For the G1 grid, the discretization and turbulence modeling uncertainty are greater than the uncertainty from either set of regression models. The turbulence modeling uncertainty grows as

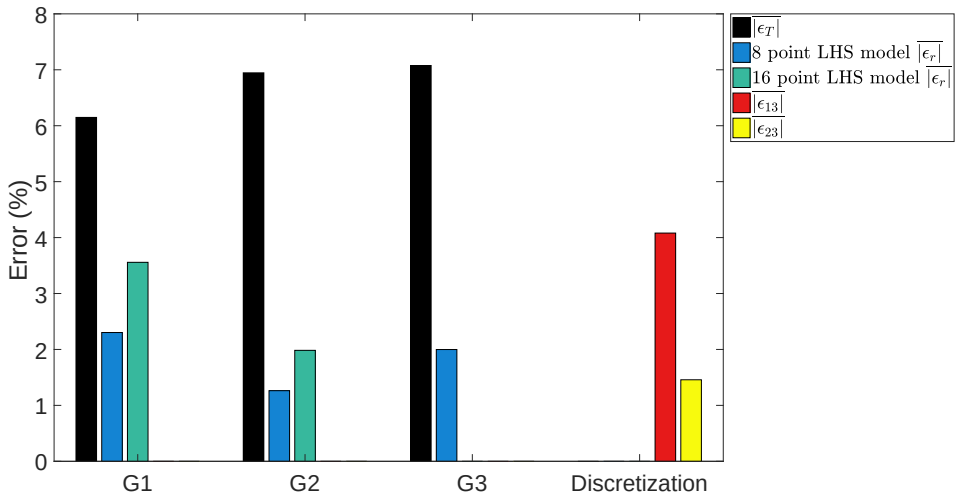

Figure 3.9: Average of the absolute value of the different error terms for calculating K_T for LHS points 1-4.

the grid is refined. For both the G2 and G3 grids, the turbulence modeling uncertainty is greater than the regression modeling uncertainty. This indicates that the regression model does a sufficient job at predicting the thrust of the propeller since when either the eight point or sixteen point set of data is used to train the model, the uncertainty of the regression model is less than the sum of the discretization and turbulence modeling uncertainty. Fig. 3.10 illustrates a similar set of features for the calculation of the torque. Fig. 3.11 and Fig. 3.12 illustrate the uncertainty in the prediction of C_{Fx} and C_{Fy} respectively. The difference in the prediction of the rudder force coefficients tends to be dominated by the turbulence modeling error, except in the case of the C_{Fy} calculation on the G1 grid, for which there is larger discrepancy between the G1 grid and the G3 grid, thus the $\overline{|\epsilon_{13}|}$ dominates. Ultimately, these figures illustrate that the regression modeling error is small relative to the uncertainty in the CFD itself and that the regression model trained with eight LHS points is sufficient to predict the propeller and rudder force coefficients with less induced uncertainty than the underlying inaccuracy as the CFD itself. Furthermore, it can be seen that either the

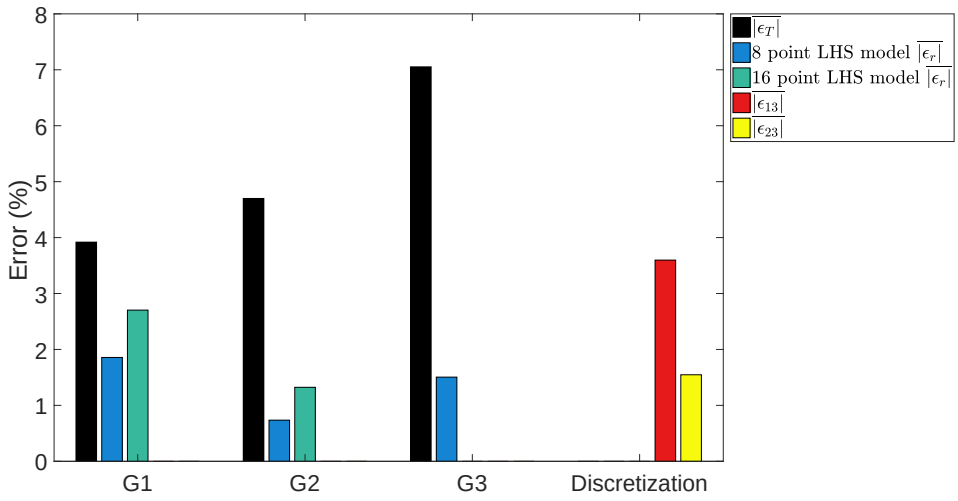

Figure 3.10: Average of the absolute value of the different error terms for calculating K_Q for LHS points 1-4.

G2 or G3 grids should be used to train the model, and that is is useful to compare the effect of using the Spalart Allmaras turbulence model and the $k-\omega$ SST turbulence model to train the model to evaluate the effect on a maneuver.

To further characterize the error, each component of the error including the discretization error, the turbulence modeling error, and the regression modeling error can be combined into an error metric $\overline{|\epsilon|}$ as defined by Eqn. 3.7. This error metric is the L_2 norm of the different components of error.

$$\overline{|\epsilon|} = \sqrt{\overline{|\epsilon_{i3}|}^2 + \overline{|\epsilon_T|}^2 + \overline{|\epsilon_r|}^2} \tag{3.7}$$

The $\overline{|\epsilon|}$ for the K_T, K_Q, C_{Fx}, C_{Fy} is 7.21%, 5.02%, 16.57%, and 19.89% respectively when the regression model is generated with the eight point training set with the G2 grid with the Spalart-Allmaras turbulence model. Similarly, the $\overline{|\epsilon|}$ for the K_T, K_Q, C_{Fx}, C_{Fy} is 7.36%, 5.12%, 16.52%, and 20.60% respectively when the regression model is generated with the sixteen point training set with the G2 grid with the Spalart-Allmaras turbulence model. The differences in $\overline{|\epsilon|}$ whether the eight point or the

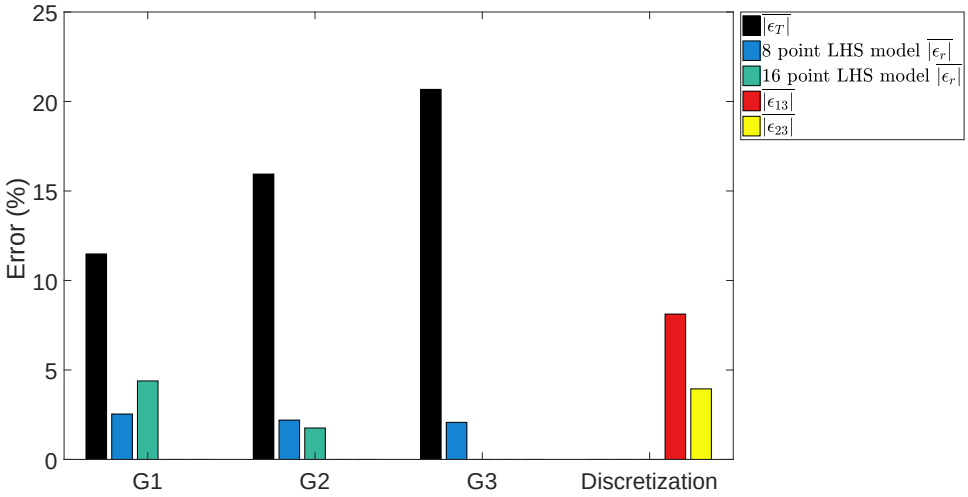

Figure 3.11: Average of the absolute value of the different error terms for calculating C_{Fx} for LHS points 1-4.

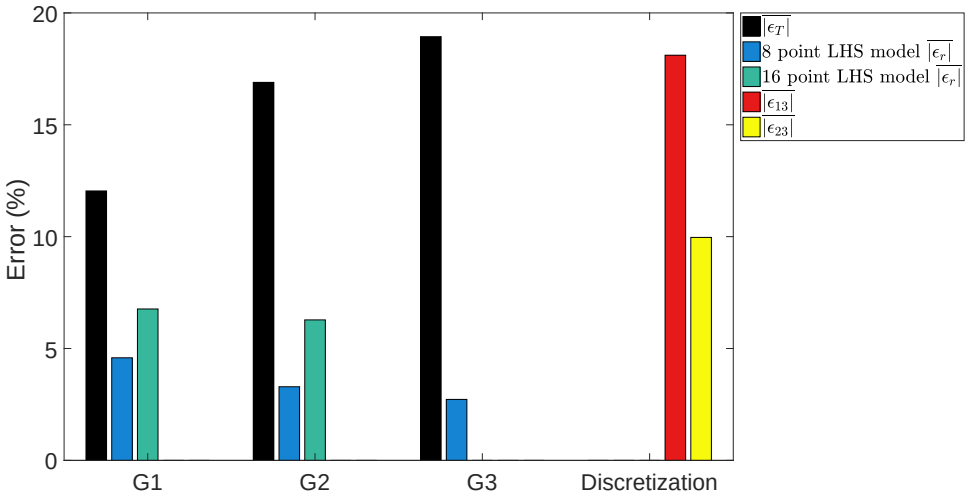

Figure 3.12: Average of the absolute value of the different error terms for calculating C_{Fy} for LHS points 1-4.

sixteen point sets is quite small. Furthermore, the $\overline{|\epsilon_r|}$ is small relative to $\overline{|\epsilon|}$.

3.2.5 Comparison Between Different Models

The difference between the different models can also be visualized in J-β space and characterized by the difference between the different models with respect to each other. First, different linear regression models are compared to each other by examining the response surface in J-β space. Subsequently, an alternative to linear regression is examined by generating response surfaces using Gaussian process regression.

Side Force Coefficient It is important to note that the K_S for this vessel is small. Therefore, the examination of the different components of the error associated with the K_S has not been emphasized in this chapter. However, the response surface for the -10K_S generated with different sets of training data can be seen in Fig. A.1. The force is scaled by a factor of ten to help illustrate the variation in the response surface and it is negative since the force is towards starboard.

Comparison Between Linear Regression Models Fig. 3.13 shows the K_T, K_Q, C_{Fx}, and C_{Fy} for the linear regression based data-driven model trained with eight LHS points with the G3 grid. The rudder force coefficients are based on the force of the complete semi-horned rudder. The K_T and K_Q are dominated by J but there is some variation due to β. The propeller is more heavily loaded at low J than high J and also tends to have higher loading at lower β values. The magnitude of the force coefficients on the rudder are greatest at high J and low β. Since the rudder is deflected it operates at a high angle of attack at this condition which causes the force on the rudder to be high. As the vessel performs a turning circle maneuver the vessel slows down reducing J and the β increases, thus, as the vessel performs the maneuver the loading on the rudder decreases. At high β and high J the rudder force coefficients become small; similarly at low β and low J the rudder force coefficients

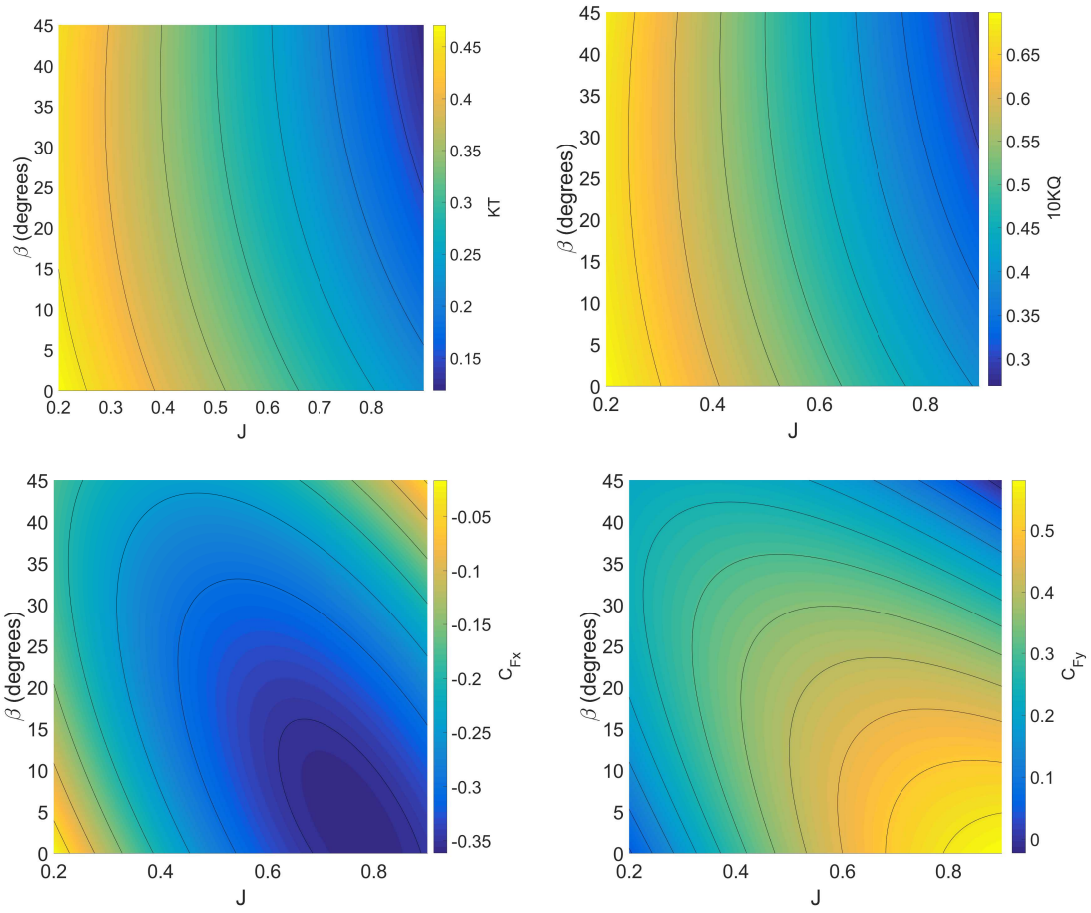

Figure 3.13: Contours of force coefficients in J-β space. K_T is shown in the upper left. K_Q is shown in the upper right. C_{Fx} is shown in the lower left. C_{Fy} is shown in the lower right.

become small, but the vessel does not operate in these regions of J-β space as shown by Fig. 3.5.

The response surfaces generated with other linear regression models can be evaluated with respect to the baseline model shown in Fig. 3.13. Fig. 3.14 shows how the response surfaces for different sets of training compare to the response surface generated with the eight point LHS set on the G3 grid for the prediction of K_T in J-β. The comparison is made by using the prediction of the model trained with the eight point set on the G3 grid M_3, subtracting the prediction of the comparison model M_i, and dividing the difference by M_3 as shown by Eqn. 3.8.

$$\epsilon_{i3} = \frac{M_3 - M_i}{M_3} \tag{3.8}$$

The response surfaces illustrated in these figures show the percent difference value in the K_T predicted with each model, with respect to the K_T value predicted by the linear regression generated with the G3 eight point LHS set. All of the cases are generated with the linear regression models trained with the double body CFD simulations that use the Spalart Allmaras turbulence model. The title of each image in the figure denotes the training data used to train the model and the grid density used to train the model. The top row of images shows the percent difference in the prediction of the two linear regression models' prediction of K_T with respect to the linear regression model generated from the eight point LHS set for the G3 grid; and the bottom row shows the difference between the linear regression models generated with the G2 versus the G3 grids. The left column shows the effect of using eight points to train the model and the column on the right shows the effect of using sixteen points. It is evident that there is more discrepancy between the G1 and G3 grids than there is between the G2 and G3 grids. This is logical since the data that is used to train the model, which was examined earlier, demonstrated that the G2 grid had less underlying discretization error and this has propagated to the data-driven model. The largest magnitude of error occurs in the lower left hand corner of the J-β space and at high values of J with large β; both of these regions are outside of the operating space of the vessel. In general there is rather low error in the J-β region that the vessel predominantly operates in during a maneuver as shown by Fig. 3.5. The form of the response surface illustrating the difference between each model in comparison to the eight point LHS G3 linear regression response surface is very similar for the calculation of K_Q and K_T as shown by Fig. A.2.

As shown earlier by Fig. 3.5, the rudder force coefficients are low in magnitude in the upper-right and lower-left corners of the J-β space considered in these figures

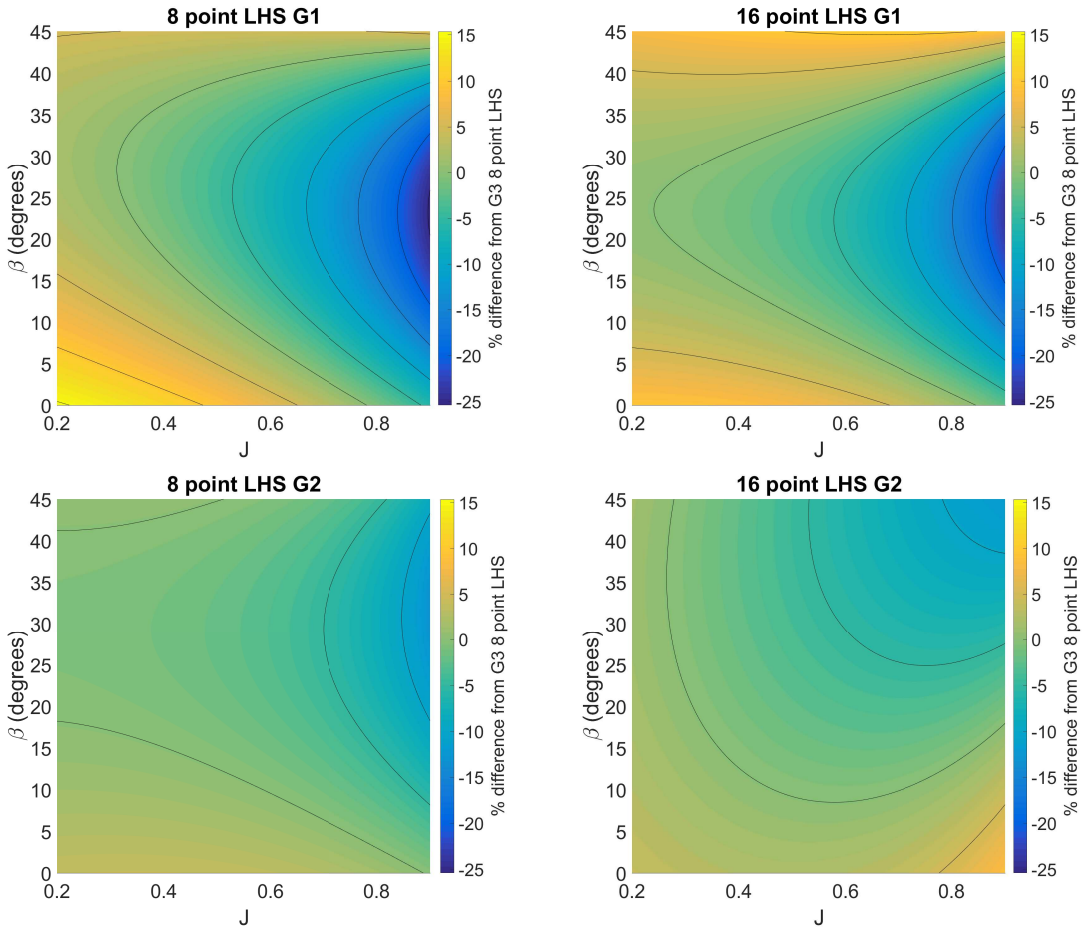

Figure 3.14: Contours of percent difference of different linear regression models' prediction of K_T with respect to the 8 point LHS G3 model in J-β space.

showing the response surfaces. For this reason, in Fig. A.3 and Fig. 3.15, this region of the plot is clipped since even small differences in the force coefficients would lead to deceivingly large amplitudes of error. These figures show the percent difference in C_{Fx} and C_{Fy} respectively for the different linear regression response surfaces with respect to the linear regression trained with the eight point G3 set of data. The deviation is largest for the models generated with the G1 grids in comparison to the G3 based model. Furthermore, the deviation is largest for high values of J and low values of β. This is sensible since the rudder is deflected to 35° and at low values of β, the rudder operates at a high angle of incidence which leads to separation. Thus, this indicates that the level of separation on the rudder is modeled differently between the grids and that this difference in underlying discretization uncertainty manifests itself in the data-driven model, which is a straightforward, yet important conclusion to draw.

Comparison between linear regression and Gaussian process regression It is important to consider the effect of utilizing different data-driven modeling approaches beyond the number of samples used and the quality of data used to train the model. Here, the linear regression model is compared to the Gaussian process regression model. For this, the rudder forces modeled are those of only the movable part of the rudder. Later in this chapter, the linear regression and Gaussian process regression predictions of the rudder force are calculated as a function of experimental motions, and only the force of the movable part of the rudder was reported by the experiments (*SIMMAN* (2020)). The models examined here use the CFD results from the G2 double-body grid with the Spalart-Allmaras turbulence model.

Two variations of the Gaussian process regression model are examined. One that assumes that the noise in the training is very small (σ_n=0.0001) and another that calculates the σ_n based upon the uncertainty of the underlying CFD used to train and

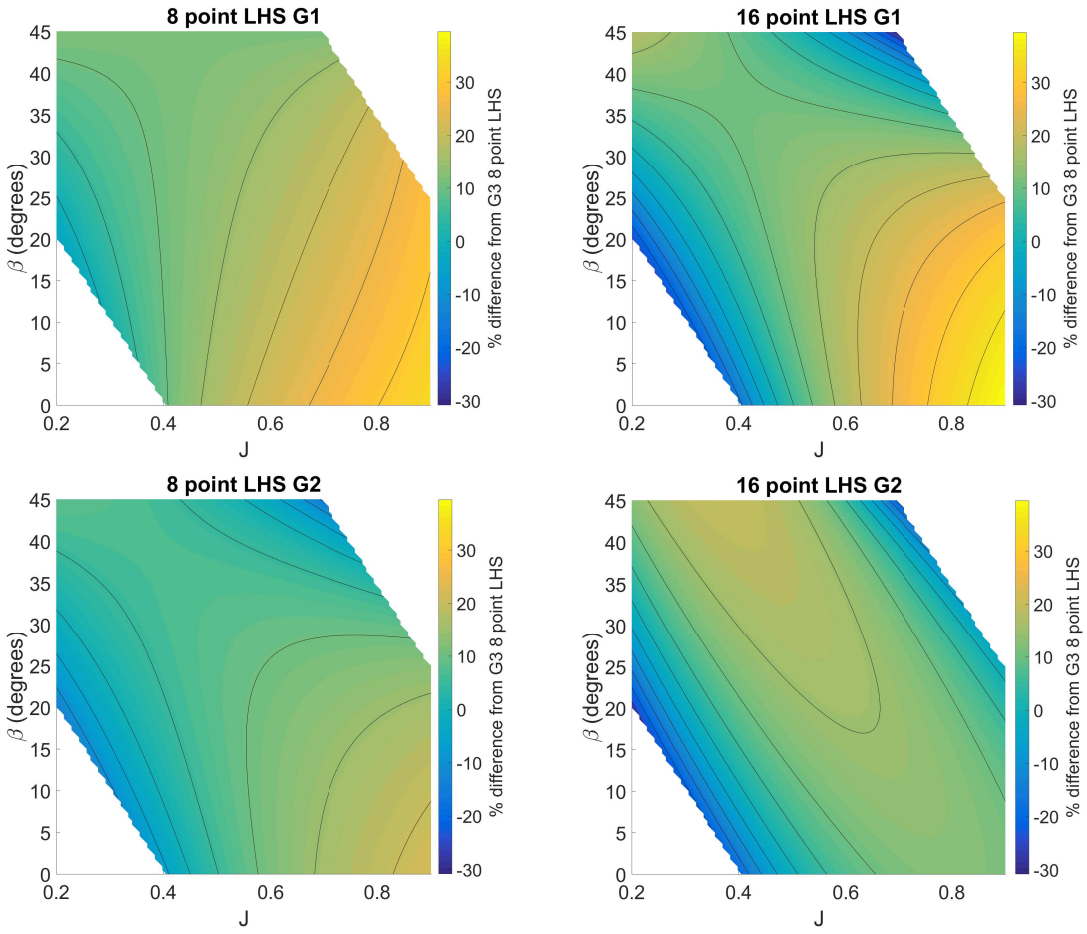

Figure 3.15: Contours of percent difference of different linear regression models' prediction of C_{Fy} with respect to the 8 point LHS G3 model in J-β space.

validate the model. The standard deviation of the uncertainty due to discretization error between the G2 and G3 grid using the Spalarat Allmaras turbulence model is used to calculate σ_{nd}. The standard deviation of the uncertainty to turbulence modeling uncertainty between the two turbulence models examined on the G2 grid for LHS points one through twelve is calculated and denoted σ_{nT}. When σ_n is calculated from the uncertainty in the underlying CFD, σ_n is calculated by the sum of σ_{nd} and σ_{nT}. When σ_n is calculated from the uncertainty in the underlying CFD.

First, the variation in the response surface between the linear regression prediction and the Gaussian process regression that assumes a low σ_n is examined. Fig. 3.16 shows the response surface prediction of K_T for the linear regression model on the left and the Gaussian process regression for a low level of σ_n=0.0001 on the right. The overall form of the response surfaces for the prediction of K_Q is very similar, so this is shown in Fig. A.4. The response surface of all the linear regression model predictions are quite smooth. On the other hand, the response surface predicted with the Gaussian process regression with a low level of noise has lots of variation, especially as training points are added for the 16 and 24 point sample sets. Both methods illustrate that the K_T and K_Q are predominantly dependent upon J, but there is some β dependence. Intuitively, the response surfaces for K_T and K_Q should be dominated by J and the response surface should be smooth. The large amounts of variation in the response surface predicted with the Gaussian process regression prediction is likely caused by over-fitting to spurious data. To elucidate this, a Gaussian process regression model is also generated for a calculated σ_n and the error of each model relative to the validation CFD data is examined once this set of response surfaces is analyzed further.

Fig. 3.17 and Fig. 3.18 show the response surfaces for the linear regression model and the Gaussian process regression model with a low σ_n for the C_{Fx} and C_{Fy} of the movable part of the rudder. Intuition dictates that the rudder loading should

be largest at low values of β when the deflected rudder operates at a high angle of incidence. Additionally, except in regions of deep stall when the flow is heavily separated, the response surface should be smooth. The response surfaces generated with the linear regression model fit this intuition, but the response surfaces generated with the Gaussian process regression are not smooth. With the Gaussian process regression models, the response surface changes as more points are added. However, one can visually see that the response surface for the linear regression model converges as more samples are used. That said, all of the models generated for these figures predict that the rudder force coefficients are largest when J is high and β. The rudder is less heavily loaded at higher values of β and lower values of J (based on the domain examined), which is where the steady state part of the turn occurs. The rudder is least heavily loaded for high J and high β, but this is outside of the operating parameter space for the vessel during a maneuver.

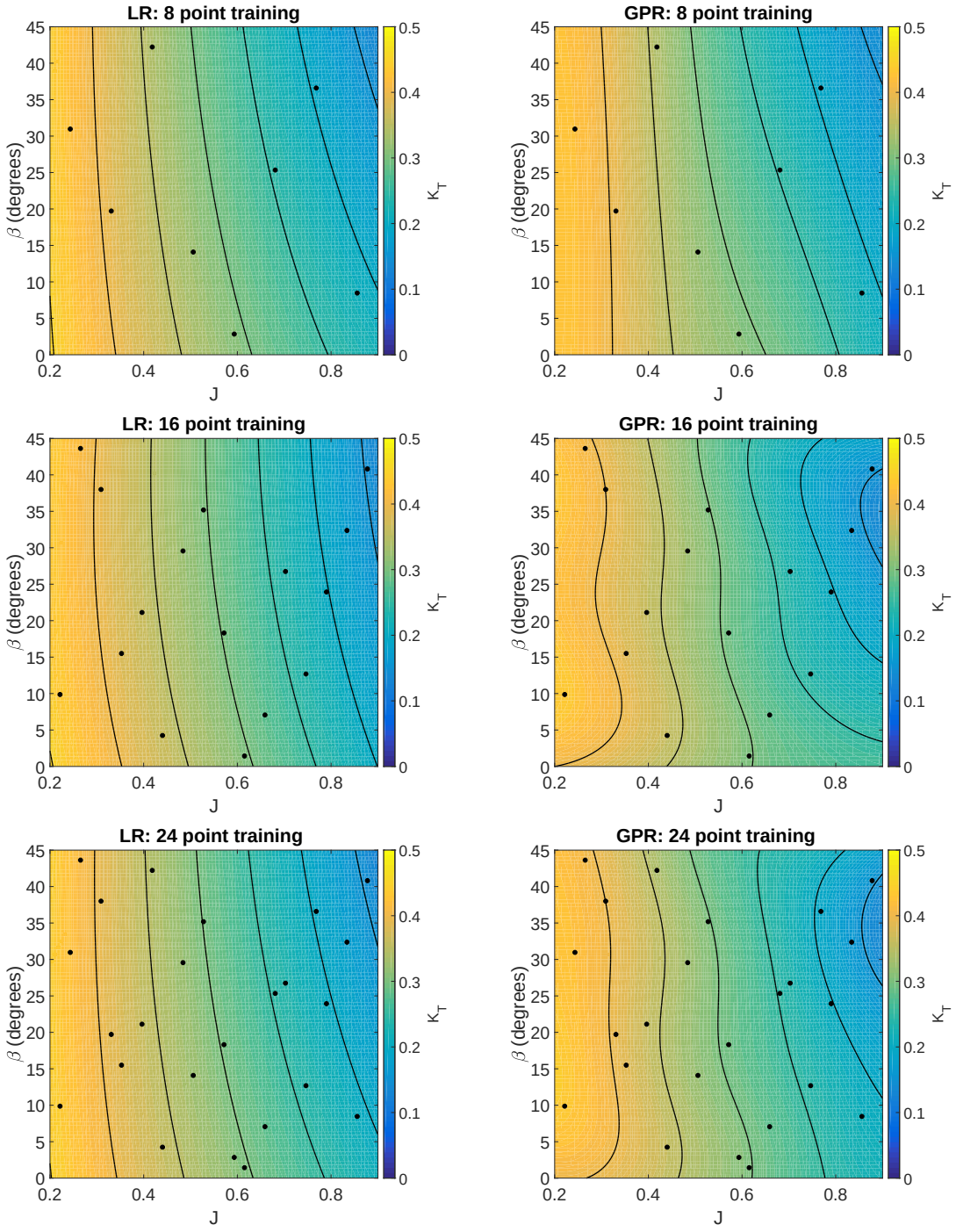

Figure 3.16: Contours of K_T in J-β space for the KCS propeller with different sets of training data, with black dots illustrating the different samples used to train the model. Linear regression response surface is shown on the left. Gaussian process regression with σ_n=0.0001 predicted response surface is shown on the right. More training points are used for each subsequent row.

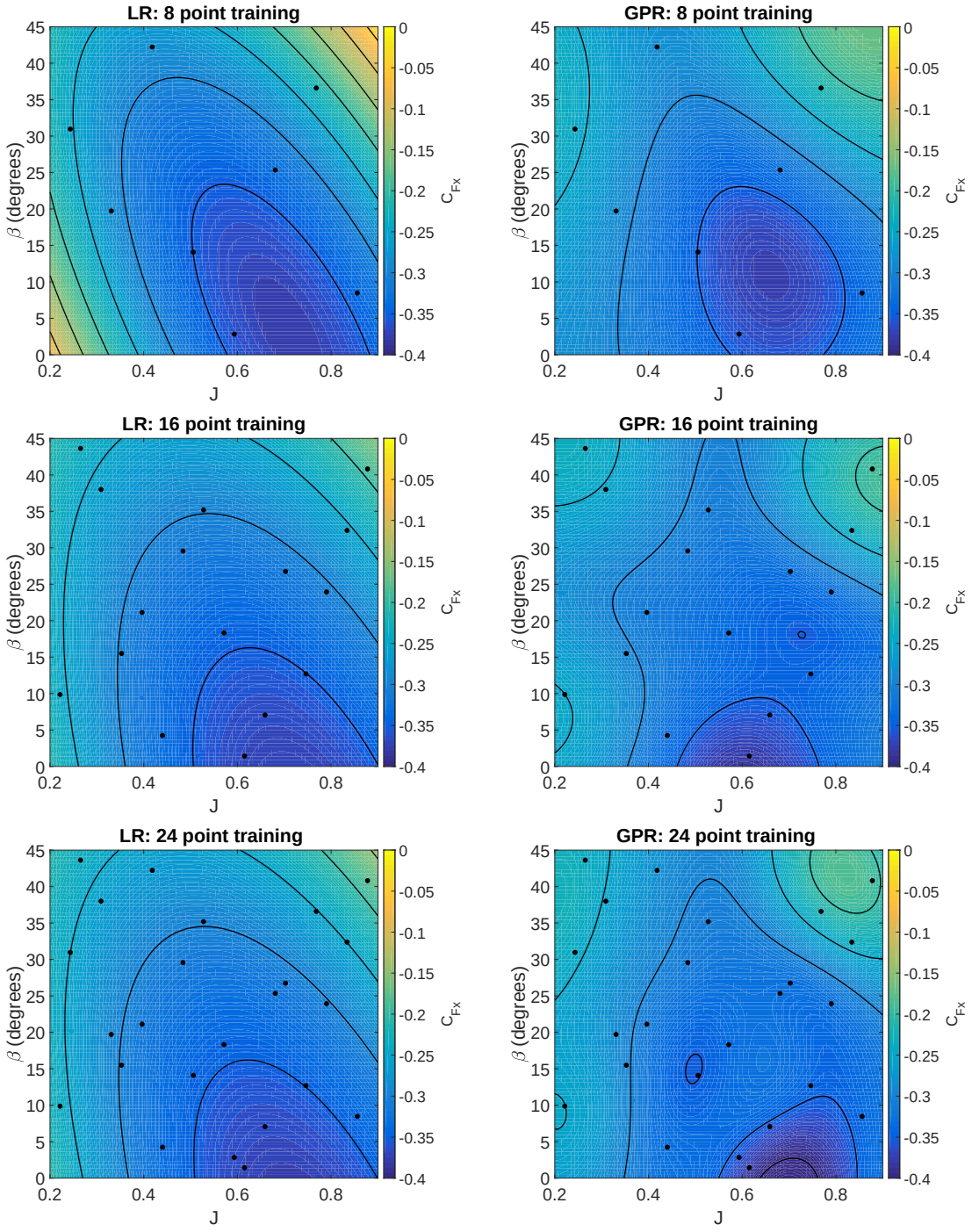

Figure 3.17: Contours of C_{Fx} in J-β space for the KCS rudder with different sets of training data, with black dots illustrating the different samples used to train the model. Linear regression response surface is shown on the left. Gaussian process regression with σ_n=0.0001 predicted response surface is shown on the right. More training points are used for each subsequent row.

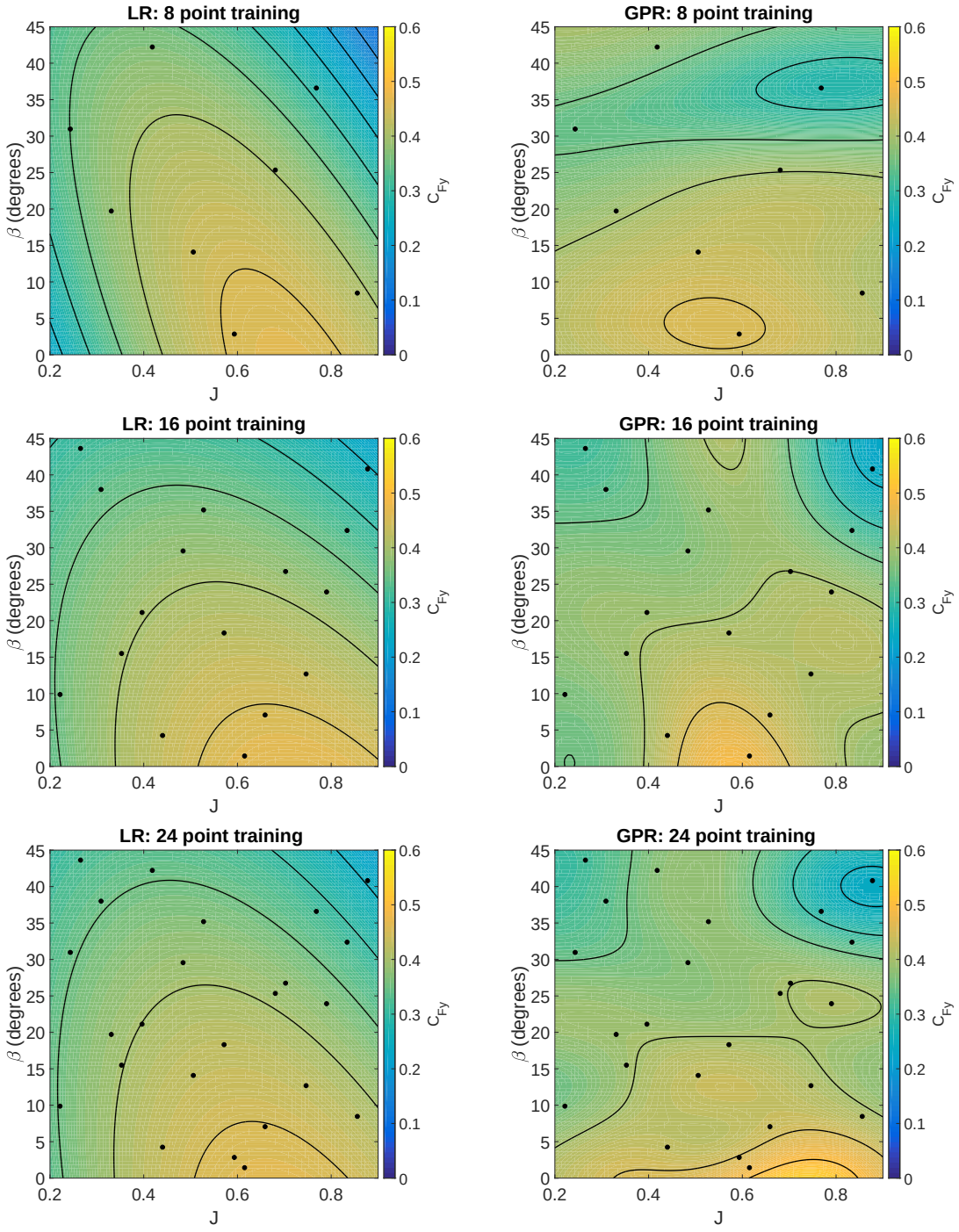

Figure 3.18: Contours of C_{Fy} in J-β space for the KCS rudder with different sets of training data, with black dots illustrating the different samples used to train the model. Linear regression response surface is shown on the left. Gaussian process regression with σ_n=0.0001 predicted response surface is shown on the right. More training points are used for each subsequent row.

An alternative to setting σ_n to a very low number is to instead use a calculated σ_n with Gaussian process regression. Accounting for the uncertainty caused by discretization error and turbulence modeling error, σ_n can be calculated as 0.024 for K_T, 0.034 for $10K_Q$, 0.046 for C_{Fx}, and 0.079 for C_{Fy}. Fig. 3.19 shows the response surfaces for K_T and C_{Fy} and Fig. A.5 shows the response surfaces for K_Q and C_{Fx}. When the inherent noise in the training data is accounted for, the curves become much smoother such that the noise is assumed to be very small. In the case of the K_T the response surface becomes very similar to that predicted with the linear regression models, especially when more sample points are added. A similar trend is shown for the prediction of K_Q. On the other hand, the C_{Fy} prediction becomes obfuscated by noise. In the case of only eight training points the C_{Fy} is almost constant and when additional training points are added there is only dependence upon β. The σ_n for the C_{Fx} is not as substantial as the σ_n for the C_{Fy}, but when the calculated σ_n for the C_{Fx} is used in conjunction with the eight point sample, the prediction is again dominated by noise. On the other hand, smooth predictions that are qualitatively similar to the linear regression model are found when more samples are used.

The reason for the discrepancy in terms of the response surface based upon the treatment of σ_n can be investigated by examining the bounds of the 95% confidence interval. In Chapter 2, the bounds of the 95% confidence interval were shown for the simple one dimensional case. Here, in two dimensions, it is most straightforward to illustrate the bounds of this 95% confidence interval range by the difference between the upper and lower bounds of the 95% confidence interval for each respective coefficient. For clarity, this value can then be nondimensionalized by the local value of the desired coefficient. Fig. 3.20 shows the uncertainty prediction interval of K_T (Δ_{KT}) as a percent value of the local value of K_T predicted by the respective Gaussian process regression model. Eqn. 3.9 shows how Δ_{KT} is calculated in terms of the upper bound of K_T provided by the GPR uncertainty K_{TU}, and the lower bound of the K_T pro-

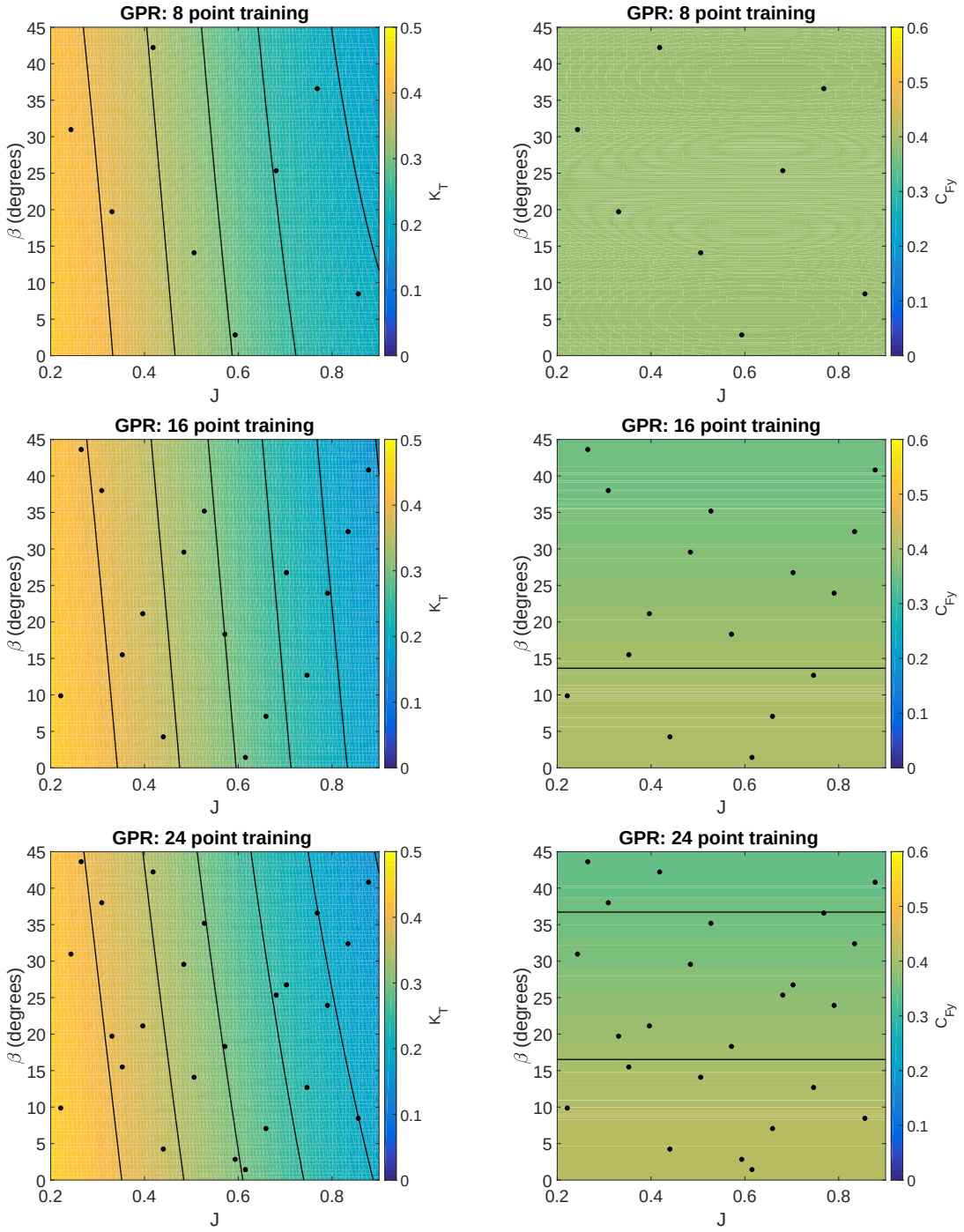

Figure 3.19: Contours of K_T and C_{Fy} in J-β space for the KCS propeller and rudder with different sets of training data, with black dots illustrating the different samples used to train the model. All results are for Gaussian process regression with a calculated σ_n. The left column of images show K_T while the right column shows C_{Fy}. More training points are used for each subsequent row.

vided by the GPR uncertainty K_{TL}. On the left of Fig. 3.20 the uncertainty interval is plotted for the low σ_n based models and on the right the calculated $\sigma_n$models are shown. This illustrates that when σ_n is assumed low the uncertainty at the training point is low. If σ_n was to be zero, then the error at each training point would be zero. On the other hand, when the expected noise in the training is accounted for, the training points cease to be treated as exact and uncertainty is present even at training points. In the case of assuming negligible noise, if the uncertainty of the training data was negligible, then the uncertainty plots shown here would indicate where additional samples should be made to improve the prediction and uncertainty of the Gaussian process regression. The plots shown here indicate that the uncertainty induced by the discretization and turbulence modeling error is significant. The percent value of this bound is especially high for high values of J when the calculated σ_n is used since the value of K_T is low for high values of J.

$$\Delta_{KT} = K_{TU} - K_{TL} \tag{3.9}$$

In the case of the K_T prediction, the expected response surface is easier to predict than the response surface for the rudder force coefficients. Additionally, it has been illustrated that the rudder force coefficients contain more discretization uncertainty and turbulence modeling uncertainty compared to the K_T. Thus, it is also useful to consider the 95% confidence interval for the C_{Fy} which is shown in Fig. 3.21. This figure illustrates that the uncertainty when σ_n is near a training point the uncertainty is near zero. As training points are added, the regions of high uncertainty decrease, whereas there are pockets of uncertainty when only eight training points are used. On the other hand, very large levels of uncertainty are present when the σ_n is calculated. When there are only eight training points used, the response surface is dominated by noise and there is no coherent response. In this case, the uncertainty is also very large, on the order of 100% of the prediction itself.

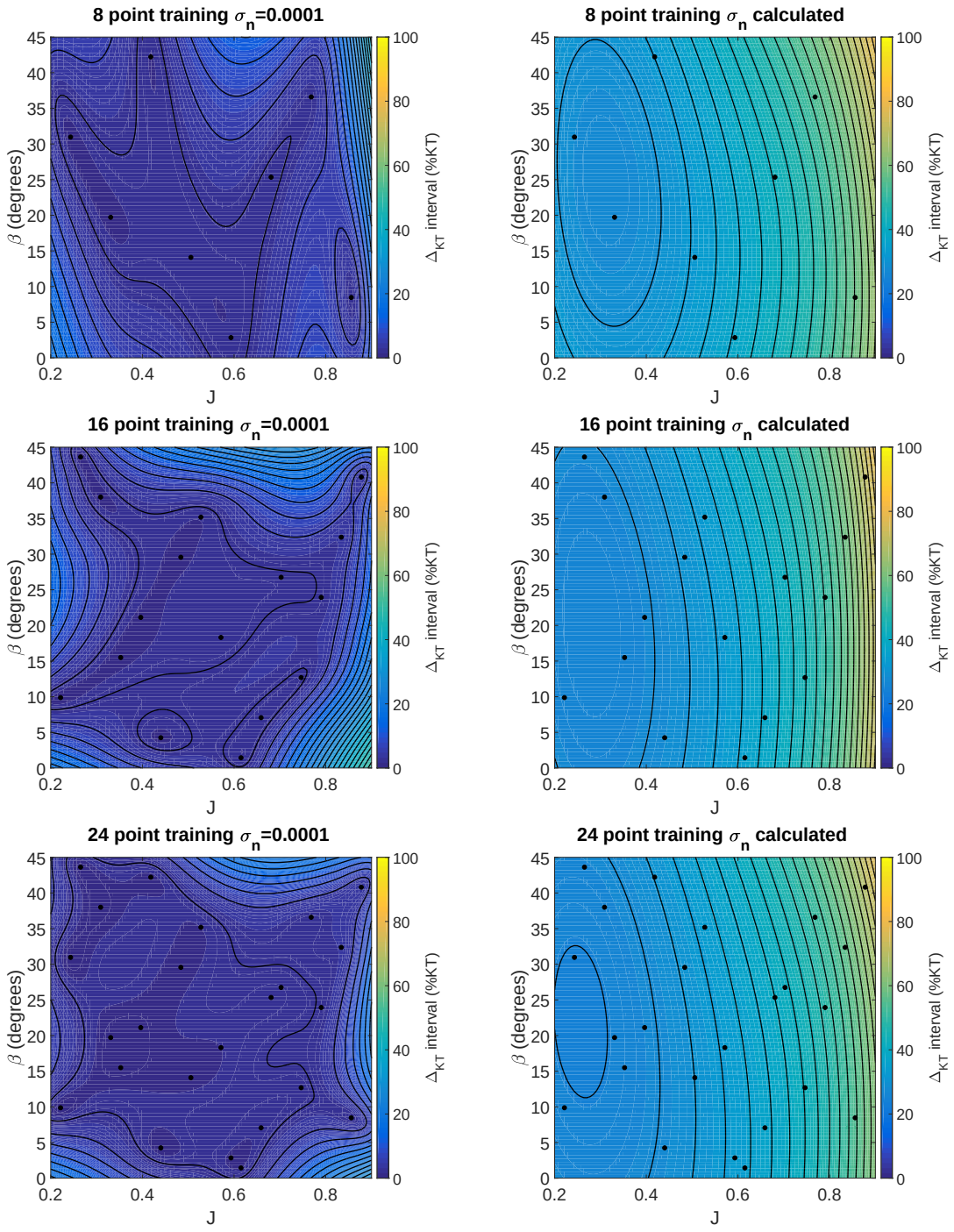

Figure 3.20: Contours of the uncertainty in the 95% confidence interval for K_T non-dimensionalized by the local value of K_T predicted by the GPR models in J-β space for the KCS propeller with different sets of training data, with black dots illustrating the different samples used to train the model. The left column show results for the low value of σ_n and the right column shows the results for calculated σ_n. More training points are used for each subsequent row.

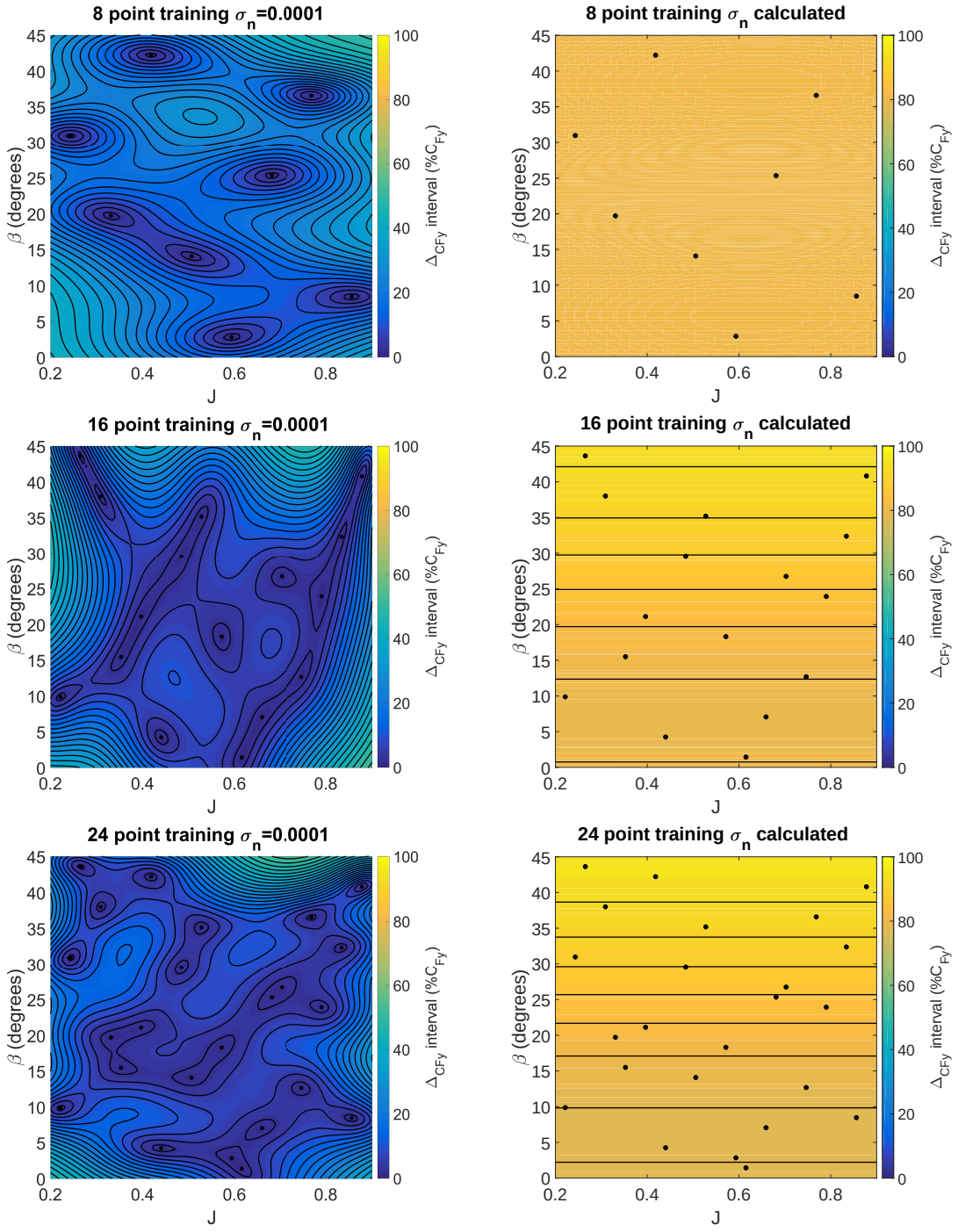

Figure 3.21: Contours of the uncertainty in the 95% confidence interval for C_{Fy} non-dimensionalized by the local value of C_{Fy} predicted by the GPR models in J-β space for the KCS rudder with different sets of training data, with black dots illustrating the different samples used to train the model. The left column show results for the low value of σ_n and the right column shows the results for calculated σ_n. More training points are used for each subsequent row.

The three different models examined (Gaussian process regression with low σ_n, Gaussian process regression with calculated σ_n, and linear regression) can be evaluated based on how well the respective models predict validation CFD data. For this the CFD results for the four point LHS set of data on the G2 grid is used. The average absolute value of the percent error between each respective model's prediction of the CFD result for each respective parameter is averaged and reported in Table 3.11. On average, across all of the parameters and all of the training sets, the linear regression models outperform the Gaussian process regression models. Also, when σ_n is very low, the prediction outperforms the equivalent model for when σ_n is calculated. Incidentally, the linear regression model with only eight training points performs the best of all of the linear regression models, but the 24 point based model outperforms the model trained with 16 points in general. This table indicates that the linear regression model is likely a better choice for training a surrogate model for sparsely sampled CFD data, which may contain inherent error. Higher fidelity CFD models would reduce the calculated σ_n, but the added work associated with implementing Gaussian process regression in a CFD package is not appealing unless there is high confidence in the predictions across the J-β space, such that the lack of smoothness in the response surface can be attributed to real effects, instead of uncertainty in the training data. Gaussian process regression could also be useful as a means to select additional points to improve the training of a linear regression model.

3.3 Calm Water Turning Circle Force Analysis

The data-driven model predictions of the propeller and rudder forces are compared to the experimental forces for the vessel performing a turning circle with the rudder deflected to 35° at an initial Froude number of 0.157. In Chapter IV, several variations of the linear regression based propeller and rudder model are applied to VOF CFD maneuvering simulations of the vessel performing the maneuver with the body force

Table 3.11: Modeling error for different Gaussian process regression models and linear regression models for K_T, K_Q, C_{Fx} on the movable part of the rudder, and C_{Fy} on the movable part of the rudder.

Training set	Parameter	GPR σ_n calculated Average % error	GPR σ_n=0.0001 Average % error	Linear regression Average % error
8 point	K_T	2.358	1.384	1.263
16 point	K_T	3.865	2.249	1.984
24 point	K_T	2.484	1.228	1.410
8 point	K_Q	2.313	0.967	0.736
16 point	K_Q	3.041	1.828	1.323
24 point	K_Q	1.797	1.019	0.918
8 point	C_{Fx}	8.223	2.667	2.567
16 point	C_{Fx}	4.385	3.654	1.956
24 point	C_{Fx}	3.543	3.300	2.354
8 point	C_{Fy}	8.213	3.880	1.764
16 point	C_{Fy}	7.635	4.070	3.360
24 point	C_{Fy}	6.812	3.050	2.803

based model.

3.3.1 Comparison of Model Forces with Calm Water Experimental Forces

The propeller thrust coefficient, propeller torque coefficient, and rudder normal force are provided in the experimental results as a function of time. Using the motions of the vessel in the experiments the model can be used to determine these force coefficients. For this, a new data-driven regression model is generated for the rudder forces since only the normal force on the moving part of the rudder, with surface area S_R of 0.008 m^2 is reported. The rudder normal force is calculated from the rudder forces calculated from C_{Fy} and from C_{Fx}.

Comparison between different linear regression models Fig. 3.22, Fig. 3.23, and Fig. 3.24 show the thrust, torque, and rudder normal force on the movable part of the rudder for different linear regression based models in comparison to the experimental forces. The linear regression models use the experimental motions of the

vessel to predict the force on the propeller and the rudder. The rudder is actuated at 0 s, but the flow has been established over the hull before this point. The propeller and rudder models neglect the rudder actuation and assume that the rudder is fully deflected. These plots illustrate the results of six different linear regression models. It is noteworthy, that then noise in the force prediction is based on the noise in the experimental motions. All of these models use the same feature vector as defined by Eqn. 2.25. The difference between the different models is the data that is used to train them. The figure legends denote the experimental data, which set of LHS points are used to train a specific model (8 point versus 16 point LHS sets), which double body grid is used to train the model (G1, G2, or G3), and which turbulence model is used. The Spalart Allmaras turbulence model is used to train all of the data-driven models shown in these figures except for the `8 point G2` $k-\omega$ `SST` case, which uses the $k-\omega$ SST turbulence model for training.

Fig. 3.22 and Fig. 3.23 illustrate that the model that uses the $k-\omega$ SST turbulence model for training produces more thrust and torque than the models that are trained with the Spalart Allmaras turbulence model. The propensity for the $k-\omega$ SST turbulence model to produce more thrust than the Spalart Allmaras turbulence model is highlighted by Table 3.5 shown earlier. This illustrates how the turbulence modeling discrepancy propagates into the data-driven model and can lead to different results. Conversely, Fig. 3.24 illustrates how the model trained with $k-\omega$ SST turbulence model produces less rudder normal force than the other data-driven models examined here.

An alternative to the feature vector that includes the features J^2 and β^2, is to use a low order feature (LOF) feature vector that includes $[1\ J\ \beta\ J\beta]$. This can help illustrate whether the original feature vector over-fits or under-fits the data and it also permits the ability to use a smaller training set to develop a model to evaluate if fewer than eight points are suitable for constructing a data-driven model. Fig. 3.25,

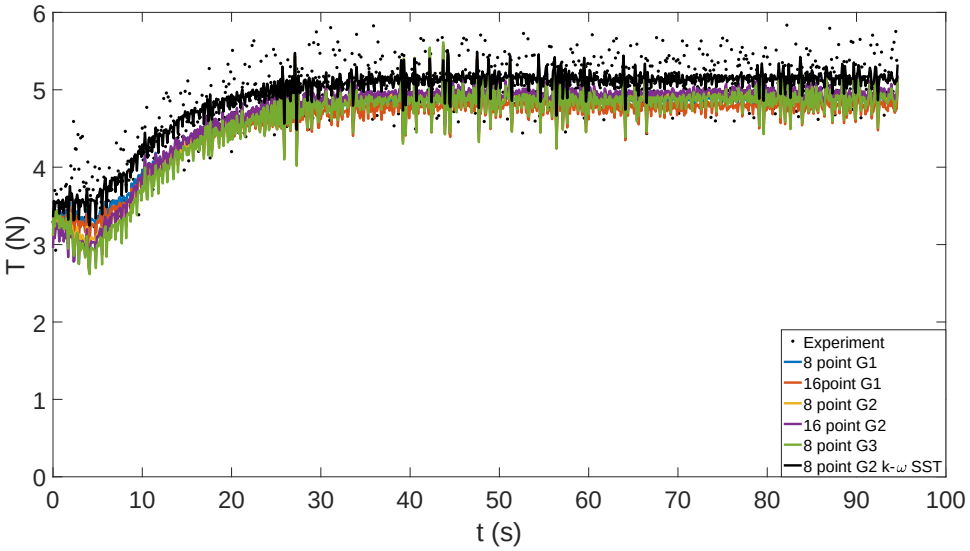

Figure 3.22: Experimental thrust compared to linear regression based data-driven thrust using motions of experiment in calm water.

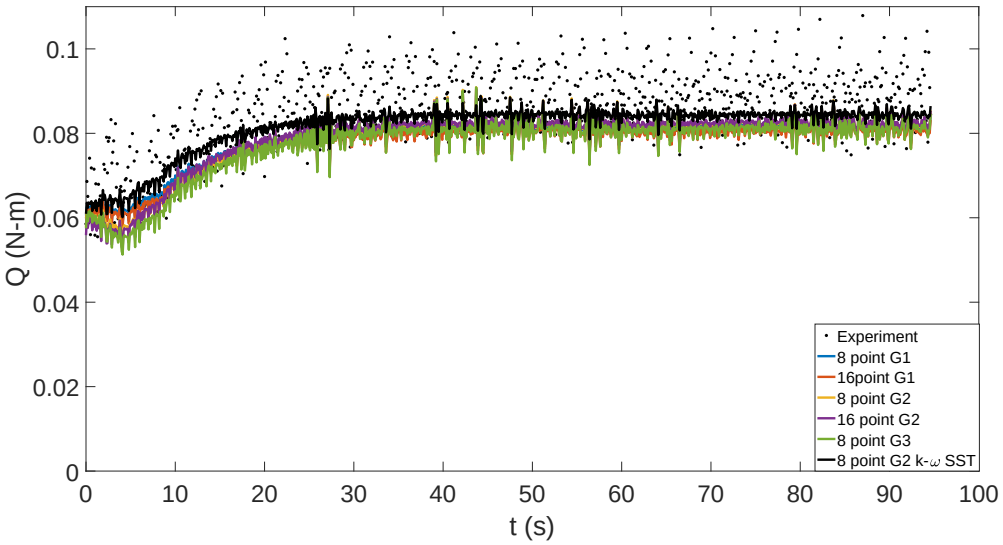

Figure 3.23: Experimental torque compared to linear regression based data-driven torque using motions of experiment in calm water.

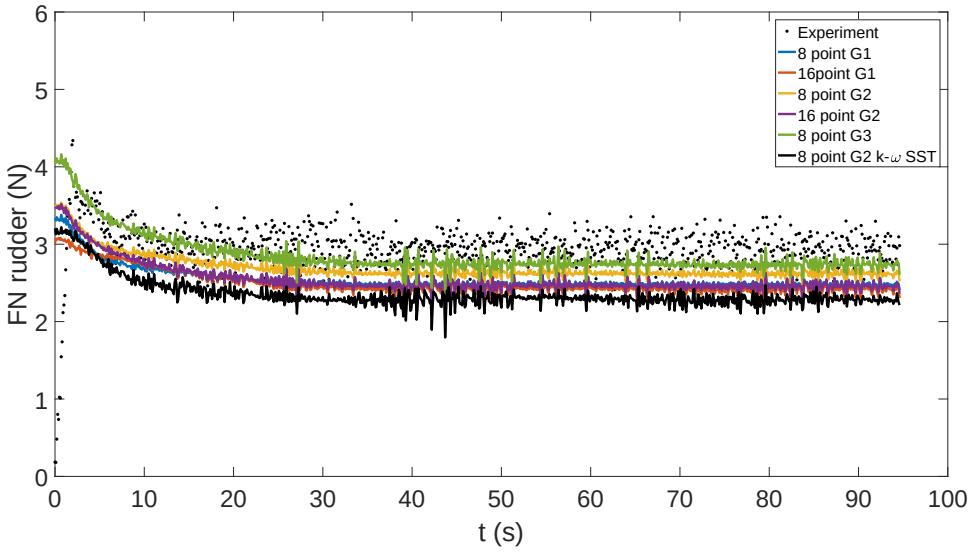

Figure 3.24: Experimental rudder normal force compared to linear regression based data-driven rudder normal force using motions of experiment in calm water.

Fig. 3.26, and Fig. 3.27 show the propeller thrust, propeller torque, and rudder normal force on the movable part of the rudder respectively for a variety of linear regression models trained with different numbers of sample points and grids using the double-body training CFD with the Spalart Allmaras turbulence model. This set of plots focuses only upon methods trained with the G2 and G3 grids, since there is less discretization uncertainty. As in the previous set of figures the eight point data set for the G2 and G3 grid is used with the original feature vector, along with the sixteen point data set on the G2 grid. The four point LHS set is also used to train a linear regression model with the low order feature vector. Furthermore, the sixteen point G2 set is used to train a low order linear regression model, along with the eight point G3 set. Fig. 3.25 and Fig. 3.26 show that there is rather good agreement between the different models and their prediction of the thrust and torque respectively with regard to each other. In the figures, the four point sets are all trained with the low order feature vector, and the denotation of `LOF` indicates that the model is trained

with the low order feature vector for models that are trained with more data points. The LOF cases tend to produce slightly more thrust than the other cases, but the differences are very small between the different cases. There are more discernible differences in the prediction of the rudder normal force with the different methods as shown by Fig. 3.27. After the rudder is actuated and before the vessel reaches the steady part of the turn there is deviation when the low order feature vector is used. For example, the sixteen point LOF G2 case calculates a level rudder normal force much earlier than the other cases. On the other hand, if only the four point LHS set is used to train the model with the G3 grid, an excessive rudder normal force is predicted at five seconds. In the steady part of the turn, most of the cases are in good agreement with each other with the model trained with the eight point LHS set on the G3 grid predicting the highest rudder normal force. This investigation illustrates that reasonable results can be predicted when only four samples are used in J-β space, but it is prudent to use more samples such as to prevent over-fitting and to gain the accuracy of including quadratic terms, since it is not expected that the response surface is purely linearly dependent upon J and β. Assuming a linear variation in terms of J and β can provide reasonable results; however, it is known that even open water propeller curves show nonlinear dependence upon J and in oblique flow when separation occurs the dependence upon β is nonlinear as well. It is also noteworthy that once the vessel reaches the steady part of the turn it has nearly constant J and β, thus, the steady forces predicted in these plots are only evaluating one point in J-β space.

Table 3.12 shows the experimental mean, standard deviation σ, and standard error for the thrust, torque, and rudder normal force. The standard error is defined by Eqn. 3.10 in terms of σ and the number of samples N.

$$\epsilon_E = \frac{\sigma}{\sqrt{N}} \tag{3.10}$$

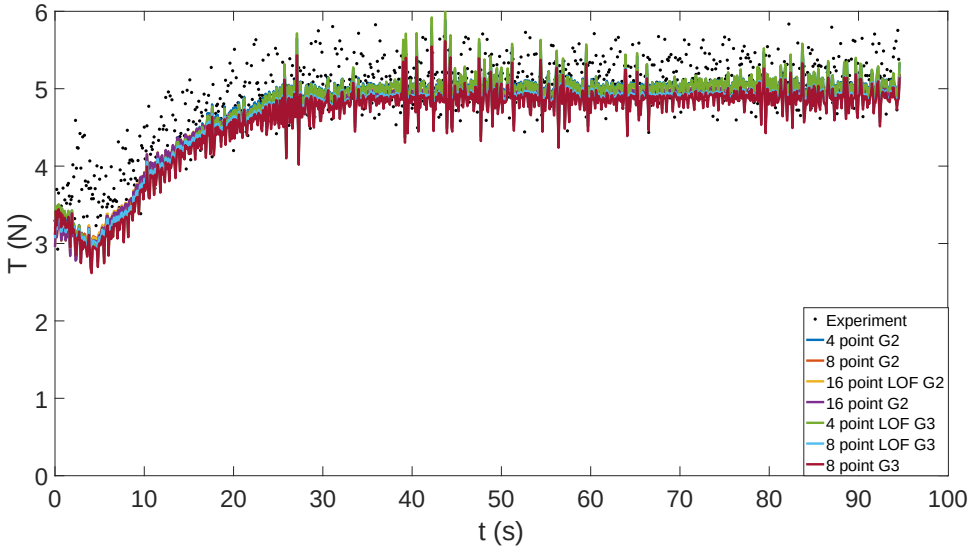

Figure 3.25: Experimental thrust compared to linear regression based data-driven thrust using motions of experiment in calm water with low order features.

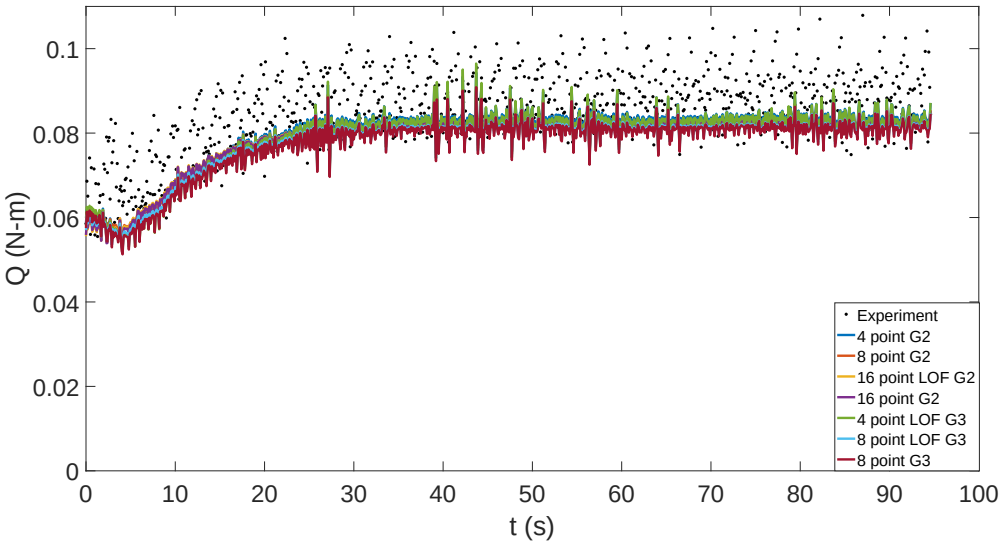

Figure 3.26: Experimental torque compared to linear regression based data-driven torque using motions of experiment in calm water with low order features.

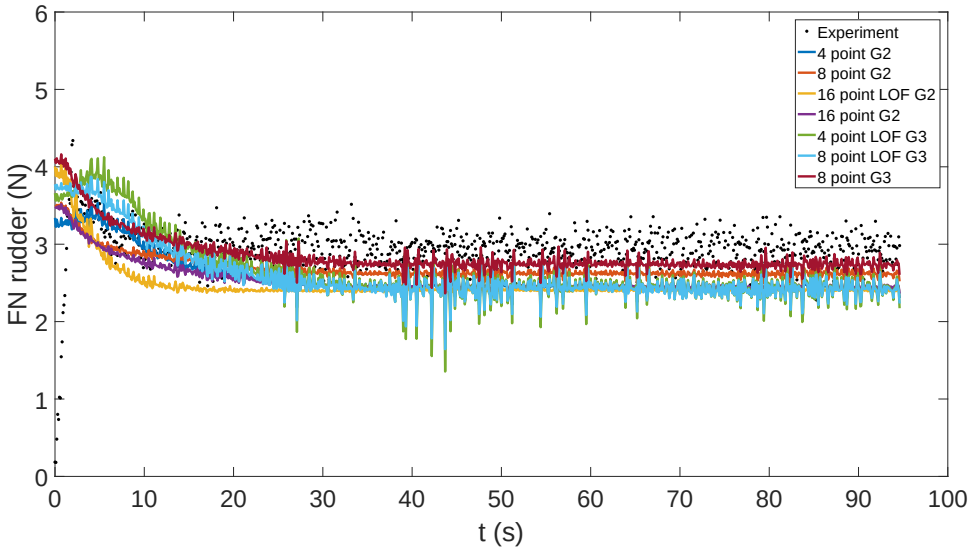

Figure 3.27: Experimental rudder normal force compared to linear regression based data-driven rudder normal force using motions of experiment in calm water with low order features.

Table 3.12: Experimental force quantities after 40 s of model scale time.

Quantity	T (N)	Q (N-m)	FN (N)
Experimental Mean	5.140	0.0883	2.923
Experimental σ	0.256	0.0062	0.179
Experimental Standard Error (ϵ_E)	0.167	0.0040	0.117

Tables 3.13-3.15 show the average thrust, torque, and rudder normal force (after 40 s) for a sample set of the linear regression models examined in Fig. 3.22 to Fig. 3.27. These tables also illustrate the error with respect to the average forces predicted by the experiment (see Table 3.12). Incidentally, it is shown that the thrust and torque predicted with four point LHS G2 based linear regression model is the most accurate relative to the experiment for the models trained with the Spalart Allmaras turbulence model. The most accurate model for the thrust and torque with regards to the experiment is the eight point LHS G2 $k-\omega$ SST based linear regression model, which has a fraction of a percent error in thrust, and is within 5% of the torque calculated in the experiment. On the other hand, the model that is trained with the

Table 3.13: Error using linear regression based data-driven model thrust predicted with experimental motions.

Model	T (N)	ϵ_m (%)	ϵ_m/ϵ_E
8 point LHS G1	4.870	5.260	1.622
16 point LHS G1	4.785	6.905	2.130
4 point LHS G2	5.040	1.948	0.601
8 point LHS G2	4.920	4.272	1.318
16 point LHS G2	4.953	3.643	1.123
4 point LHS G3	5.025	2.231	0.688
8 point LHS G3	4.876	5.135	1.584
8 point LHS G2 $k-\omega$ SST	5.143	-0.055	-0.017

Table 3.14: Error using linear regression based data-driven model torque predicted with experimental motions.

Model	Q (N-m)	ϵ_m (%)	ϵ_m/ϵ_E
8 point LHS G1	0.0815	7.671	1.674
16 point LHS G1	0.0804	8.881	1.938
4 point LHS G2	0.0834	5.508	1.202
8 point LHS G2	0.0821	7.040	1.536
16 point LHS G2	0.0823	6.804	1.485
4 point LHS G3	0.0829	6.121	1.336
8 point LHS G3	0.0810	8.234	1.797
8 point LHS G2 $k-\omega$ SST	0.0844	4.418	0.964

$k-\omega$ SST based data under-predicts the experimental rudder normal force by over 20%, whereas the best performing case is the eight point LHS G3 model.

Table 3.15: Error using linear regression based data-driven model rudder normal force predicted with experimental motions.

Model	FN (N)	ϵ_m (%)	ϵ_m/ϵ_E
8 point LHS G1	2.482	15.175	3.804
16 point LHS G1	2.424	17.172	4.304
4 point LHS G2	2.437	16.714	4.190
8 point LHS G2	2.618	10.544	2.643
16 point LHS G2	2.459	15.963	4.001
4 point LHS G3	2.436	16.767	4.203
8 point LHS G3	2.738	6.440	1.614
8 point LHS G2 $k-\omega$ SST	2.287	21.840	5.474

Comparison between linear regression and Gaussian process regression It is also important to consider if the method used to train a data-driven model significantly alters the prediction. This can be evaluated by testing different data-driven methods (linear regression and Gaussian process regression) that are developed with the same training data. Here, the Spalart-Allmaras turbulence model results on the G2 grid are used to compare the accuracy of Gaussian process regression based models and linear regression based models. Earlier, the response surfaces generated with the different methods were compared. The Gaussian process regression based model generates a response surface with lots of variation if the σ_n is assumed low, but when σ_n is calculated a smoother response surface is generated. Fig. 3.28, Fig. 3.29, and Fig. 3.30 show the predictions of thrust, torque, and rudder normal force for the different models in comparison to the experimental results. The experimental motions are used as input for the data-driven models to calculate the respective forces. The linear regression model, the Gaussian process regression model with a low σ_n, and the Gaussian process regression model with a calculated σ_n are applied. Each respective model is trained with three different training sets: the eight point LHS, the sixteen point LHS, and a combination of the eight point and sixteen point sets. The response surfaces for these models are earlier. The discrepancy between the different models in the prediction of thrust and torque as a function of time is quite small.

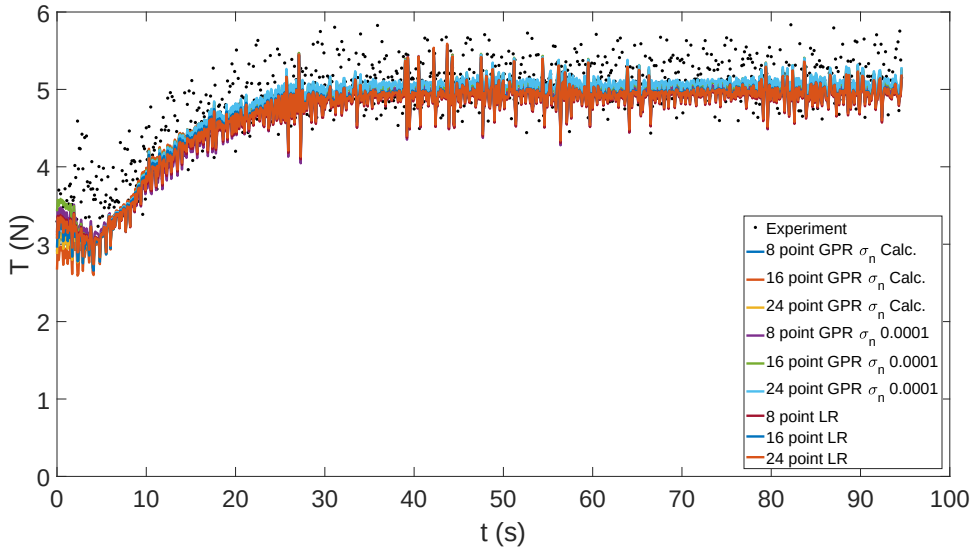

Figure 3.28: Experimental thrust compared to data-driven thrust using motions of experiment in calm water of linear regression models (LR) and a Gaussian process regression models (GPR) trained with various sample sets of the double-body G2 grid simulations with the Spalart-Allmaras turbulence model.

In terms of the calculation of the rudder normal force, the eight point set of data used to train the linear regression model and the Gaussian process regression model with a calculated σ_n lead to a force prediction that is larger than the other cases. However, the discrepancy between the different models shown here is less than the discrepancy illustrated for different sets of training data for which discretization and turbulence modeling uncertainty is demonstrated. Thus, using either linear regression or Gaussian process regression leads to a very similar prediction of the force, such that the differences between the models is rather small relative to the underlying uncertainty of the data with which they were trained.

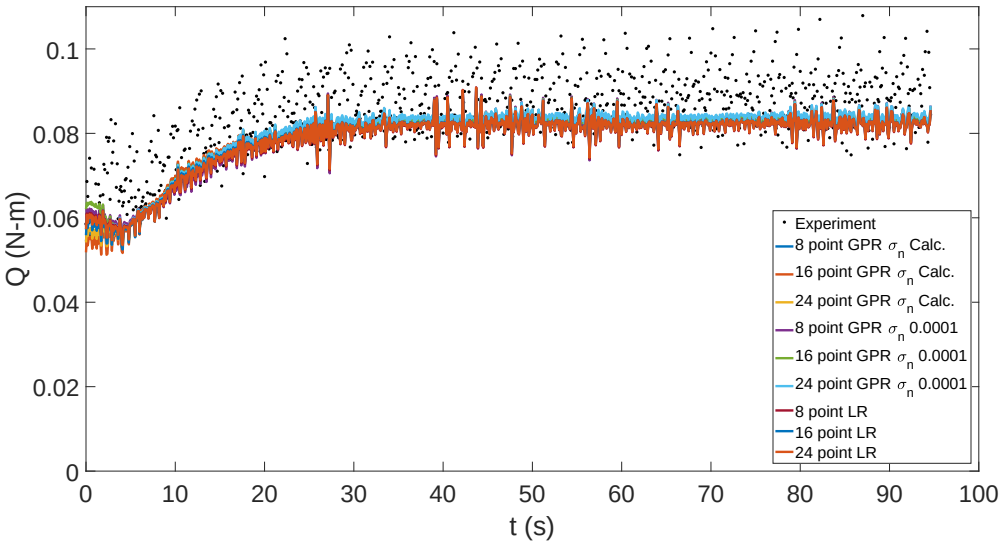

Figure 3.29: Experimental torque compared to data-driven thrust using motions of experiment in calm water of linear regression models (LR) and a Gaussian process regression models (GPR) trained with various sample sets of the double-body G2 grid simulations with the Spalart-Allmaras turbulence model.

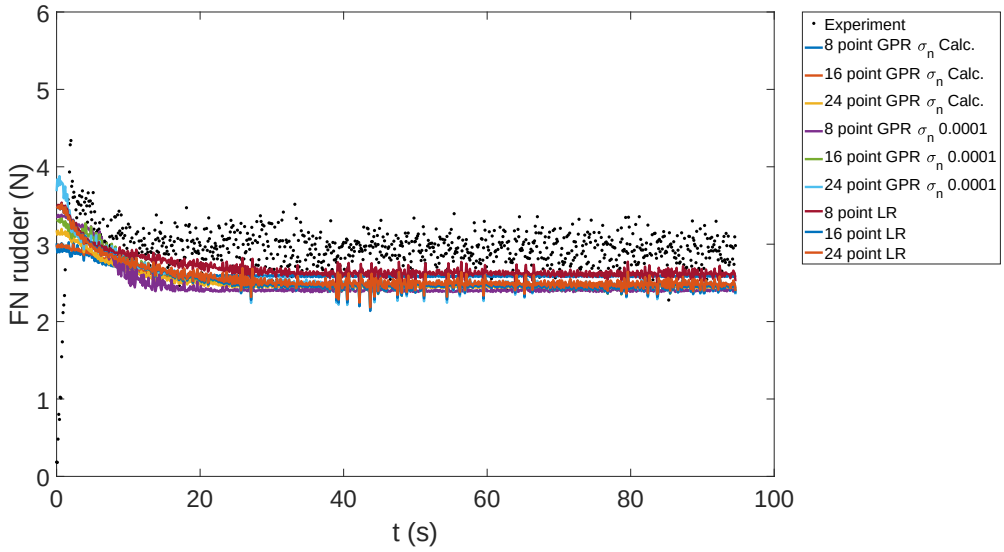

Figure 3.30: Experimental rudder normal force compared to data-driven thrust using motions of experiment in calm water of linear regression models (LR) and a Gaussian process regression models (GPR) trained with various sample sets of the double-body G2 grid simulations with the Spalart-Allmaras turbulence model.

Since Gaussian process regression is more complicated to implement in a CFD package and there is not a demonstrable benefit to Gaussian process regression over linear regression, the linear regression method is implemented in the CFD package to apply the data-driven propeller and rudder force for the VOF CFD maneuvering calculations in Chapter IV.

CHAPTER IV

Maneuvering of the KRISO Container Ship (KCS)

The turning circle characteristics of the KCS are examined in calm water and in waves for a δ of 35°. Different linear regression based data-driven propeller and rudder models are examined. The data-driven models are trained with different sets of training data and the effects on the maneuvering characteristics of the vessel are examined.

In this chapter the VOF CFD model is discussed first. Following the discussion on the maneuvering CFD setup the calm water turning circle characteristics are discussed. The methods are extended to analyzing the vessel operating in regular waves. For the discussion on the performance in regular waves, the data-driven propeller model force predictions of the propeller and rudder are first compared to the experimental force predictions. Following this, different linear regression based models are applied to the VOF CFD maneuvering simulations. The effect of using different sets of data to train the data-driven model is discussed in terms of how the maneuvering characteristics change based on the quantity of training data and the method used to generate the training data.

4.1 VOF CFD Model Development for KCS

The VOF CFD simulations are used to analyze the vessel maneuvering both in calm water and in waves. Both VOF grid setups are constructed in a similar way as the double-body grid with the main exception that the air domain is also modeled. Additionally, the propeller and rudder geometry are removed from the simulation and are instead modeled with the data-driven model. The body force of the propeller is applied to the flow via a body force region which is the cylinder that encompasses the swept volume of the propeller. The data-driven propeller and rudder model determines the forces of the propeller and the rudder based on the instantaneous motions of the vessel determined by a customized six-degree of freedom motion solver. The flow chart illustrating how the data-driven propeller and rudder forces are calculated and implemented is shown in Fig. 4.1. The model determines the thrust, side force, and torque of the propeller, as well as the yaw moment contribution to the equations of motion due to the propeller side force. Additionally, the model determines the rudder side force, resistance force, and yaw moment. Note that for the calculation of the yaw moment, the side force of the propeller and the rudder is positive towards the port side and that d and d_r are positive scalar values. These forces are applied to the equations of motion of the body and to the fluid. The body-force of the propeller thrust and torque are applied in the swept volume of the propeller according to the distribution specified by *Hoekstra* (2006), with the assigned coefficients of the model correlating to the typical open-water propeller distribution specified by *Hoekstra* (2006). The body force contribution of the side force is applied with the same distribution as the thrust, but in the orthogonal direction aligned with the direction of the side force. The body force of the rudder is neglected.

A customized six degree of freedom rigid body motion solver is used for each simulation. The vessel is accelerated from rest to the specified initial forward speed of 0.86 m/s over eight seconds of model scale time. During this ramp the vessel is free

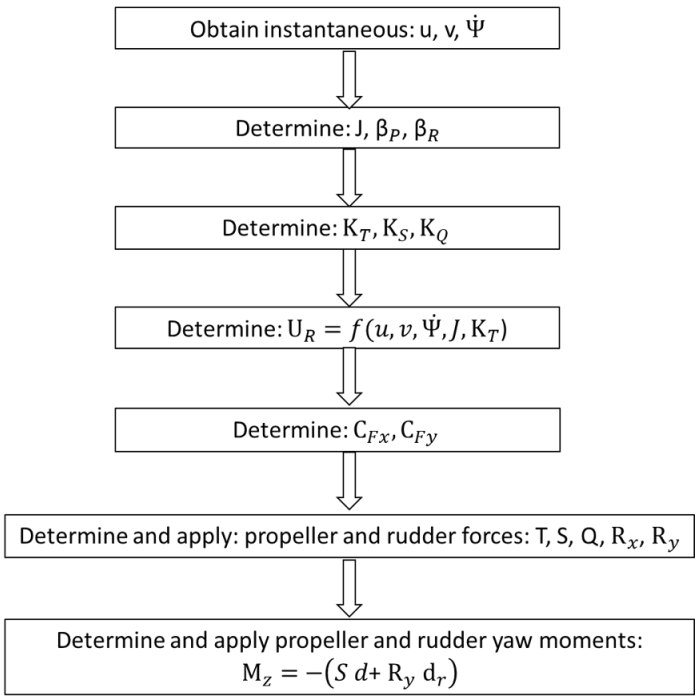

Figure 4.1: Flow chart for the implementation of the data-driven propeller and rudder model for KCS.

to pitch and heave. Once the vessel has obtained the desired speed, the speed is held constant for four seconds before it is released, and all six degrees of freedom are solved for. The data-driven propeller and rudder model apply the body force to the flow and the force and moments of the propeller and rudder to the equations of motion to perform the desired maneuvers. In this study, the force predicted by the data-driven model is calculated assuming the rudder angle is deflected to the maximum deflection angle for the duration of the simulation. In the experiment the rudder deflection rate was 20.1° per second, thus it takes 1.74 s for the propeller to deflect from the undeflected position to the maximum deflected position. This transient is ignored in this study, and instead when the CFD results are compared to the experiment the time and position are adjusted by 0.87 seconds, or the time that it would take for the propeller to reach its midway deflection point. Alternatively, the forces could be interpolated between the prediction of the data-driven model of the force with

a deflected rudder versus a calculation of the propeller and rudder forces with the rudder undeflected.

The domain for all of the VOF CFD simulations is 12.4 m long (4 L_{PP}), 3.1 m deep below the calm waterplane and 1.55 m above the calm waterplane. There are small differences in terms of the CFD domain setup between modeling the vessel in calm water and in waves. For the calm water CFD cases the domain extends 4.65 m in each lateral direction whereas for VOF CFD simulations in waves, the domain extends 6.2 m laterally, such that the domain is square when viewed from above. The vessel is at the center of the domain longitudinally and laterally and is floating on the waterplane. The boundary condition on the hull is non-slip wall modeled with wall functions. The Spalart Allmaras turbulence model is used. For the calm water CFD simulations, the domain surfaces upstream, port, and at the bottom of the domain are wave velocity inlets with zero gradient pressure. The surfaces downstream and starboard are wave velocity inlets with fixed flux pressure for the calm water case; while the surface downstream for the case in ambient waves is instead modeled with a velocity inlet with zero gradient pressure. These slight differences in boundary conditions do not result in demonstrable differences in the maneuvering characteristics. The top of the domain is a pressure inlet-outlet for all cases.

The grid setup is very similar to the double-body grids with the main difference being that the air region is also modeled. The grid densities examined are based off of the G1 and G2 grid densities for the double-body CFD simulations. One modification is that the region of the waterplane, which extends above and below the waterplane by 0.2 m, is also refined for some of the cases in the Z direction by a factor of $\sqrt{2}$ or by a factor of 2. This enables better resolution of the free surface while not significantly altering the overall grid count and driving up computational cost. The square domain used for the VOF CFD in waves uses only one grid density which is the G1 grid with

Table 4.1: Grid size for KCS VOF simulations.

Grid	Grid size (million cells)	Δ_g (m)	Δ_{gx} (m)	Δ_{gy} (m)	Δ_{gz} (m)
Calm G1	2.20	0.029	0.028	0.029	0.029
Calm G1 $\Delta z/\sqrt{2}$	2.34	0.026	0.028	0.029	0.020
Calm G1 $\Delta z/2$	2.44	0.024	0.028	0.029	0.014
Calm G2	5.78	0.020	0.020	0.020	0.020
Waves G1 $\Delta z/2$	2.96	0.024	0.028	0.029	0.014

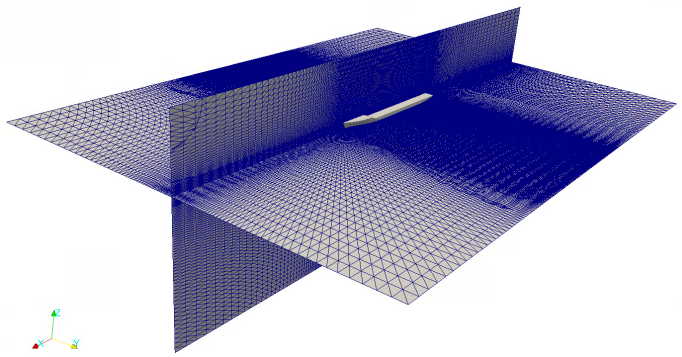

Figure 4.2: KCS G1 $\Delta z/2$ Calm Water VOF grid.

the waterplane region refined by a factor of 2. Fig. 4.2 shows the VOF grid used for the calm water CFD simulations with G1 grid density with the waterplane region refined by a factor of two. Table 4.1 shows the number of cells used for each grid as well as the discretization length in each direction in the region around the hull, extending +/− 1.86 m fore and aft from center of the domain, 0.32 m laterally in each direction, and from 0.2 m below the waterplane to 0.2 m above the waterplane. The discretization lengths in this region denoted Δ_{gx}, Δ_{gy}, and Δ_{gz} for the x, y, and z directions respectively. SnappyHexMesh is used to generate the mesh and applies one level of refinement to the whole hull. In regions of high curvature, two levels of refinement are used. The hull also has five prism layers applied with a thickness of 0.3 times the dimension of neighboring cell in the normal direction to the face.

A customized waves2Foam (*Jacobsen et al.* (2011)) package is used for the VOF simulations. In the CFD simulations for the vessel in waves a wave velocity inlet

boundary condition is applied on all of the domain surfaces except for the top where the pressure inlet-outlet is specified. For calm water cases the relaxation zones are rectangular and extend laterally and longitudinally along the perimeter of the domain except for a square in the center of the domain with length 2 L_{PP}. A potentialCurrent wave type, which specifies uniform velocity along the vertical extent of the boundary (*Jacobsen et al.* (2011)), is used with zero velocity for rectangular relaxation zones that extend to the exterior boundaries. For cases with ambient waves, the waves are specified as regular fifth order Stokes waves with a cylindrical wave relaxation zone specified with an inner radius of 1 L_{PP} from the center of gravity of the vessel, and an outer radius of the maximum extent of the domain.

4.2 CFD Maneuvering Simulations in Calm Water

The vessel is modeled with the data-driven propeller and rudder model in calm water and compared to the *SIMMAN* (2020) results. The vessel trajectory, dimensionless forward velocity, and sway velocity are examined. The forward velocity is made dimensionless by dividing the instantaneous forward velocity by the initial forward velocity of the vessel u_o which is 0.86 m/s. The vessel is examined with VOF grids with different mesh density, different temporal discretizations, and with different sets of training data for generating the data-driven propeller and rudder model. The results shown here demonstrate an improvement over initial applications of this method which used a linear regression based model to predict the forces of only the propeller, and used a simple body force distribution in conjunction with a simplified spade rudder to allow actuation of the rudder with a rotating sliding mesh (*Piro et al.* (2020), *White* (2020)).

4.2.1 Effects of Spatial Discretization

Four different spatial discretizations are examined using a baseline time step of approximately 0.008 s. The different VOF grid mesh discretizations examined include the G1 VOF grid, the G1 $\Delta z/\sqrt{2}$ VOF grid, the G1 $\Delta z/2$ VOF grid, and the G2 VOF grid. The G1 $\Delta z/\sqrt{2}$ VOF grid and the G1 $\Delta z/2$ VOF grid offer a means of adding extra refinement to the waterplane without significantly increasing the mesh count. The linear regression based propeller and rudder model is trained with the eight point LHS double-body G2 training set for all cases. This highlights a distinct advantage of using the data-driven propeller and rudder model, since the data-driven model can be trained with a rather fine grid, but it can then be applied to VOF grids that are of coarser mesh density. This is a convenient way to reduce the computational cost, since not only does the propeller and rudder model allow for a significant improvement in time step, but it also allows for the use of coarser VOF grids that do not require the fine spatial discretization around the propeller and rudder.

Fig. 4.3 shows the trajectories once the rudder is actuated for the different vessels. The rudder is deflected at the origin and the vessel begins to turn. Table 4.2 shows the maximum advance A_M, the maximum transfer T_M, and the turning circle diameter D_T for the prediction using each grid and for the experiment. The maximum advance is the maximum distance that the vessel travels in the X direction after the rudder is deflected. The maximum transfer is the maximum distance that the vessel travels in $|Y|$. The turning circle diameter is the diameter of the circle that the trajectory of the vessel makes when viewed from above during the steady part of the turn. The maximum advance (maximum value of X) is predicted within 4.2% for all of the cases relative to the experimental value. The largest deviation from the experimental prediction of the maximum advance is for the G2 grid, while the best agreement with the experiment is the G1 $\Delta z/2$ VOF grid, which has less than 1% error relative to the experiment. The maximum transfer, which is the maximum distance that the

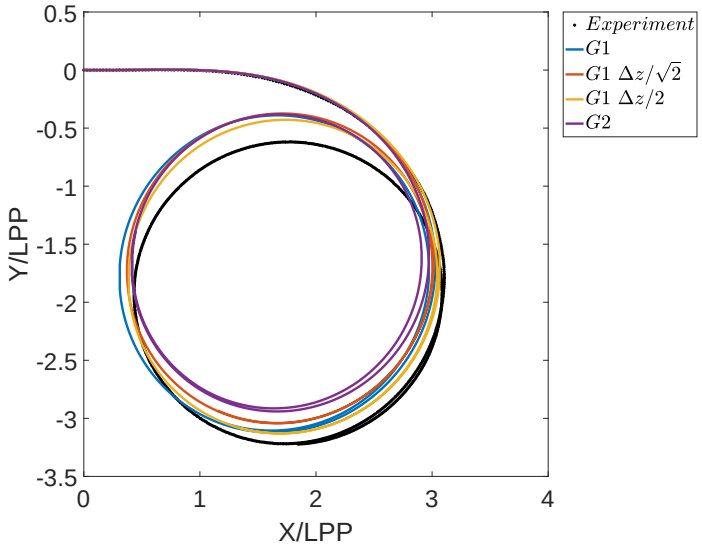

Figure 4.3: KCS turning circle maneuver trajectories with different VOF mesh discretizations using the 8 point LHS G2 training set for the data-driven propeller and rudder model.

vessel travels in $|Y|$, is predicted within 9% for all cases. Again, the G2 grid agrees least with the experimental results with an 8.7% under-prediction with respect to the experimental results. The best agreement occurs for the G1 $\Delta z/2$ grid with an under-prediction of only 2.8% with respect to the experimental results. The steady turning diameter is calculated within 5% for all cases relative to the experimental result. The experimental turning circle diameter is 2.6 L_{PP}, and the G1 grid predicts a diameter that is 4.9% larger, while the G2 grid predicts a turning circle diameter that is 1.7% tighter. Overall, the G2 grid predicts a tighter turn than the other grids. Refining the waterplane alone also leads to oscillatory convergence. For example, the G1 and the G1 $\Delta z/2$ grids both have larger maximum transfers and maximum advances than the G1 $\Delta z/\sqrt{2}$ VOF grid.

Fig. 4.4 and Fig. 4.5 show the dimensionless forward velocity and the sway velocity respectively for the different spatial discretization cases relative to the experiment. All of the CFD results under-predict the experimental forward velocity and sway

Table 4.2: Dependence of KCS turning circle characteristics on spatial discretization.

	Experiment	G1	G1 $\Delta z/\sqrt{(2)}$	$\Delta z/2$	G2
Maximum Advance (A_M/L_{PP})	3.1	3.03	3.02	3.07	2.97
Maximum Transfer (T_M/L_{PP})	3.22	3.12	3.04	3.13	2.94
Diameter (D_T/L_{PP})	2.602	2.73	2.67	2.70	2.56

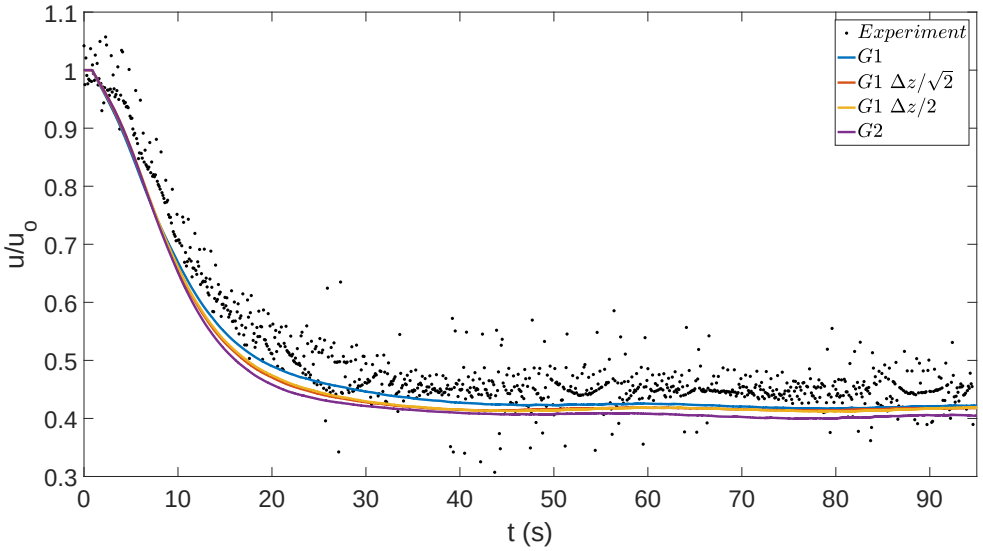

Figure 4.4: KCS turning circle maneuver dimensionless forward velocity with different VOF mesh discretizations using the 8 point LHS G2 training set for the data-driven propeller and rudder model.

velocity. Both the G1 $\Delta z/\sqrt{2}$ grid and the G1 $\Delta z/2$ grid lead to very similar forward velocities. The G2 grid has approximately 3% less forward velocity and the G1 grid has 1% more forward velocity in the steady part of the turn with respect to the G1 $\Delta z/\sqrt{2}$ grid and the G1 $\Delta z/2$ grid. In terms of sway velocity, the G1 grid has the least sway velocity whereas when the waterplane is better refined the sway velocity is larger and is in better agreement with the experiments. The fact that the forward speed is under-predicted is logical since the thrust produced by the data-driven propeller model is less than the thrust measured in the the experiment.

Overall, the G1 $\Delta z/2$ VOF grid produces reasonable results with regards to the experiments and offers a good balance between computational cost and accuracy.

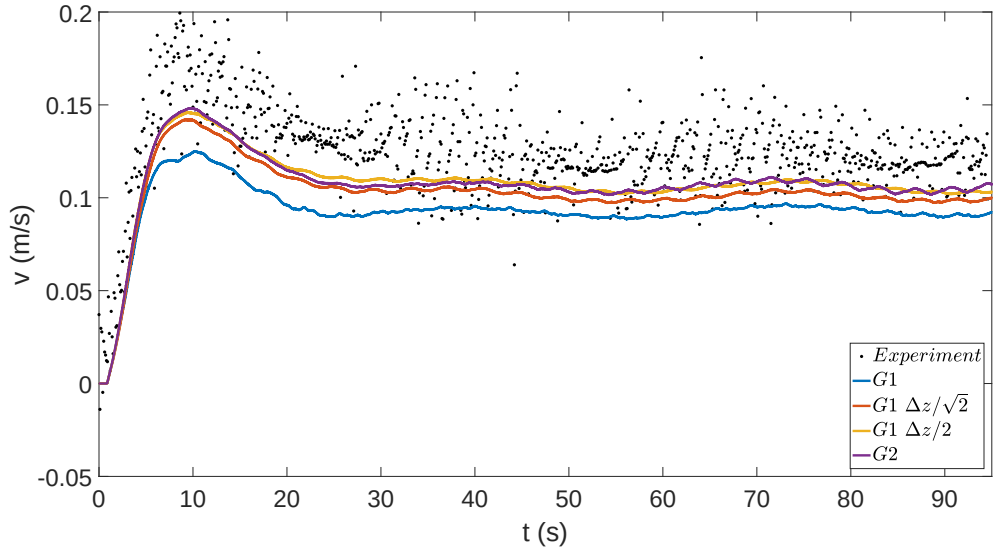

Figure 4.5: KCS turning circle maneuver sway velocity with different VOF mesh discretizations using the 8 point LHS G2 training set for the data-driven propeller and rudder model.

4.2.2 Effects of Temporal Discretization

Four different temporal discretizations are examined using the G1 Δz/2 VOF grid with the eight point LHS G2 training set with a linear regression model for the data-driven propeller and rudder model. The allowable time step is controlled by increasing the maximum allowable Courant number. The second-order backwards in time scheme is used for the time integration. Each simulation is run for 120 s, during which the first seconds ramp the velocity from rest to the initial full speed, after which the vessel is held at that constant forward speed for four seconds until the vessel is released to maneuver. Thus, the vessel performs the turning circle maneuver for 108 s which is sufficient to perform over one and a half circles. The temporal discretizations are referred to as $\Delta t/2$, the `base` case, $2\Delta t$, and $3\Delta t$; the average time step size for each of these cases over the course of the simulation is 0.004 s, 0.008 s, 0.016 s, and 0.025 s. Thus, four time steps are evaluated, the `base` time step size Δt_o, and multiples of the time step [0.5, 1, 2, 3]. These time step sizes can also be

compared to the amount that the propeller would rotate with each of these time steps while rotating at a rate of n=10.4 rps. The equivalent propeller rotation is 15°, 30°, 60°, and 94° per time step for the $\Delta t/2$, `base`, $2\Delta t$, and $3\Delta t$ cases respectively. This leads to a substantial improvement in computational cost if any of these temporal discretizations are used compared to using a discretized propeller which would rotate approximately 1° per time step. The simulations are run with 120 cores with 2.2-GHz processors. The number of cores required for the $\Delta t/2$, `base`, $2\Delta t$, and $3\Delta t$ cases are 3691 core hours, 1845 core hours, 948 core hours, and 607 core hours respectively. Despite the fact that significant computation gains can be made by allowing very large time steps, it is important to characterize the accuracy of the turning circle maneuvering characteristics to ensure that the accuracy does not degrade.

In general, the $\Delta t/2$, the `base` case, and the $2\Delta t$ cases are in good agreement with each other, but the $3\Delta t$ case deviates. Fig. 4.6 shows the turning circle trajectory for the different temporal discretizations and the experimental results. All of the cases predict a maximum advance within 2% of the experiment, however, the $3\Delta t$ case has 4.2% less of a maximum transfer; whereas the baseline temporal discretization has 2.8% less of a maximum transfer with respect to the experiment. The steady state turning circle diameter of the $3\Delta t$ case actually agrees best with the experiment and is 2.5% greater than the experiment, whereas the other temporal discretizations predict 3.3%, 3.8% and 3.8% for the $\Delta t/2$, the `base` case and the $2\Delta t$ cases respectively. Fig. 4.7 shows the maximum transfer and the turning circle diameter for the different temporal discretizations in relation to the experimental results.

The dimensionless forward velocity and the sway velocity can also be compared for the different temporal discretizations in Fig. 4.8 and Fig. 4.9 respectively. Fig. 4.8 shows that all of the CFD predictions under-predict the mean experimental forward velocity, but all of the CFD simulations except the $3\Delta t$ case have very good agreement with each other. The $3\Delta t$ case under-predicts the forward velocity of the other CFD

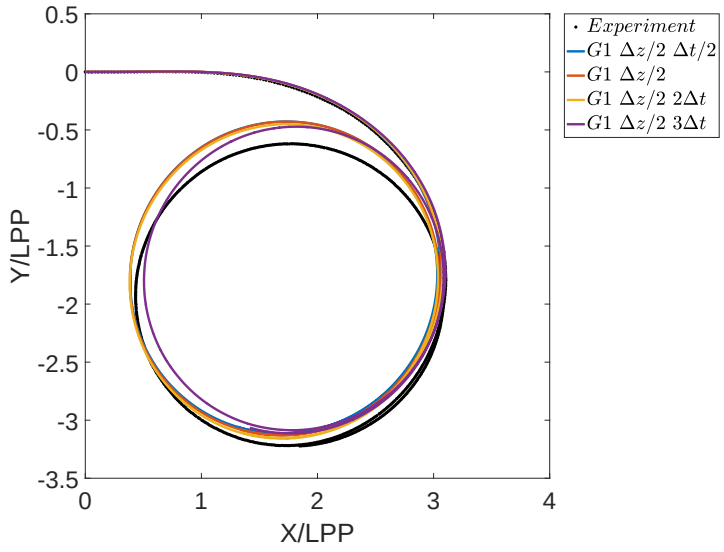

Figure 4.6: KCS turning circle maneuver trajectories with different VOF temporal discretizations using the G1 Δz/2 VOF grid with the 8 point LHS G2 training set for the data-driven propeller and rudder model.

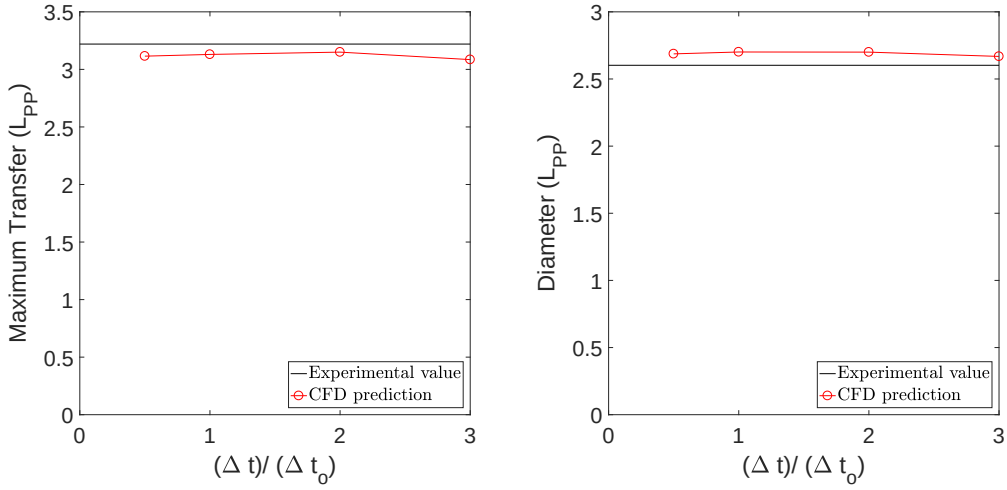

Figure 4.7: KCS turning circle maneuvering characteristics with different VOF temporal discretizations using the G1 Δz/2 VOF grid with the 8 point LHS G2 training set for the data-driven propeller and rudder model. Left: Maximum transfer. Right: Turning circle diameter.

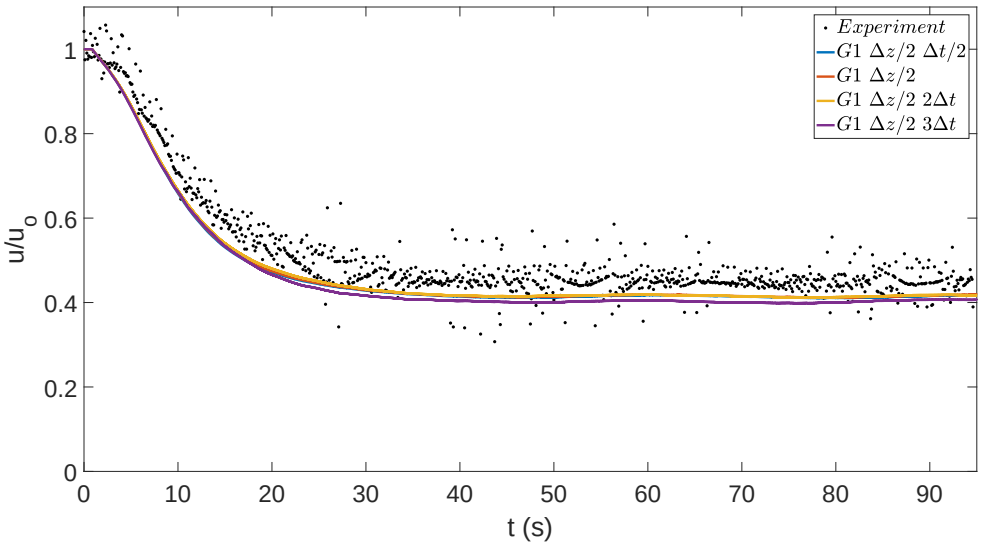

Figure 4.8: KCS turning circle maneuver dimensionless forward velocity with different VOF temporal discretizations using the G1 Δz/2 VOF grid and the 8 point LHS G2 training set for the data-driven propeller and rudder model.

cases in the steady part of the turn by approximately 3%. Fig. 4.9 shows that all of the CFD simulation under-predict the experimental sway velocity, but all of the CFD results are in good agreement with each other except the $3\Delta t$ case which over-predicts the sway velocity of the other CFD cases by almost 2%.

These results indicate that for this grid, a time step correlating to 15°, 30°, and 60° of propeller rotation lead to similar maneuvering characteristics and represent a significant improvement in the allowable time step compared to what would be required if the propeller was discretized. It is also noticeable that if very large time steps are used, then the maneuvering characteristics of the vessel are affected.

4.2.3 Effects of Different Linear Regression Based Models

It has been shown that a data-driven propeller model allows for computationally efficient modeling of the vessel performing a maneuver. It has also been demonstrated in Chapter III that the uncertainty associated with the accuracy of the data-driven

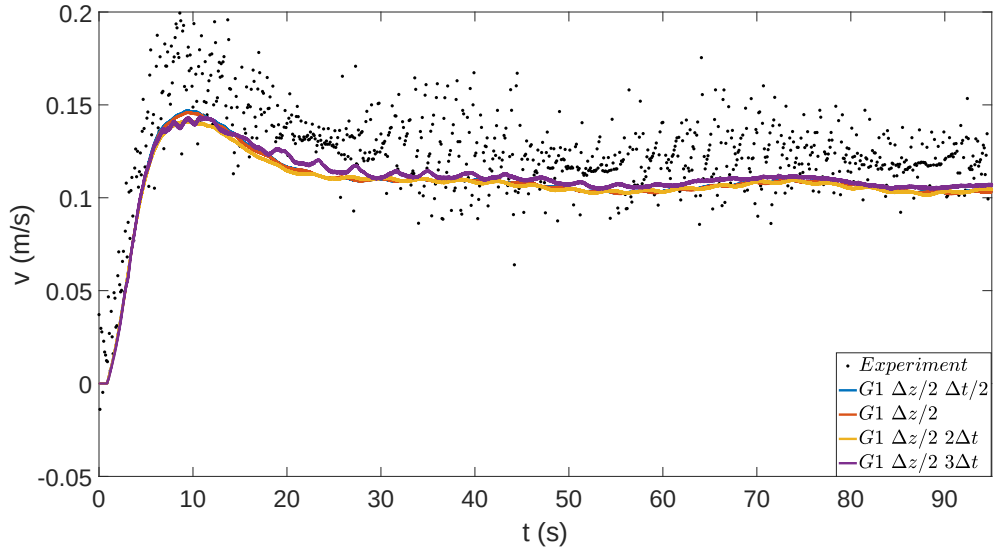

Figure 4.9: KCS turning circle maneuver sway velocity with different VOF temporal discretizations using the G1 Δz/2 VOF grid with the 8 point LHS G2 training set for the data-driven propeller and rudder model.

propeller and rudder model is heavily dependent upon the uncertainty of the data that it is trained with. The grid density used to train the data-driven model and the turbulence model used to train the model have significant effects on both the propeller and rudder forces. To help quantify the difference in the effect of the using different data-driven models, several linear regression based models are examined here.

The linear regression model that is trained with the eight point LHS training set, which used the Spalart Allmaras turbulence model, on the G2 grid is used and in the following figures the results are denoted by `G2`. Earlier when the experimental motions of the vessel were used to predict the thrust of the propeller it was found that this data-driven model slightly underestimated the thrust calculated in the experiment. For this reason, the effect of modifying the propeller revolution rate is examined. By increasing the propeller revolution rate, the propeller thrust increases. Thus, instead of applying a propeller revolution rate that matches the experimental propeller revolution rate (n=10.2 rps), variations are also applied which use an n of 10.6 rps

Table 4.3: Dependence of KCS turning circle characteristics on data-driven model.

	Maximum Advance (A_M/L_{PP})	Maximum Transfer (T_M/L_{PP})	Diameter (D_T/L_{PP})
Experiment	3.10	3.22	2.60
G2	3.07	3.13	2.70
G2 $k-\omega$ SST	3.21	3.35	2.93
G2 n=10.6 rps	3.05	3.12	2.70
G2 n=10.8 rps	3.03	3.11	2.70
G3	2.93	3.00	2.60

and 10.8 rps, denoted `G2 n = 10.6` and `G2 n = 10.8` respectively in the following figures. On the other hand, it was found earlier that the linear regression model trained with the eight point LHS set on the G2 grid with the $k-\omega$ SST turbulence model, denoted `G2` $k-\omega$ `SST`, slightly over-predicted the experimental thrust and also under-predicted the rudder normal force relative to other data-driven models. For this reason it is important to consider the effect that using a different turbulence model has on the turning circle characteristics. As has been illustrated throughout Chapter III, the differences between these models predominantly is based upon how the training data is generated, and most of the uncertainty present would be present even if the propeller was discretized for the maneuvering simulation. The final data-driven model examined is the linear regression model trained with the eight point LHS set of data using the Spalart Allmaras turbulence model on the G3 grid, denoted `G3`. This model was shown in Chapter III to produce more rudder force than the other models. The linear regression coefficients for three different sets of training data are shown in Tables A.9-A.11 for reference.

Fig. 4.10, Fig. 4.11, and Fig. 4.12 show the turning circle trajectory, the dimensionless forward velocity, and the sway velocity respectively for these different cases run with the baseline time step on the G1 $\Delta z/2$ VOF grid. Table 4.3 shows the turning circle maneuvering characteristics for the different cases as well as the experimental results for reference.

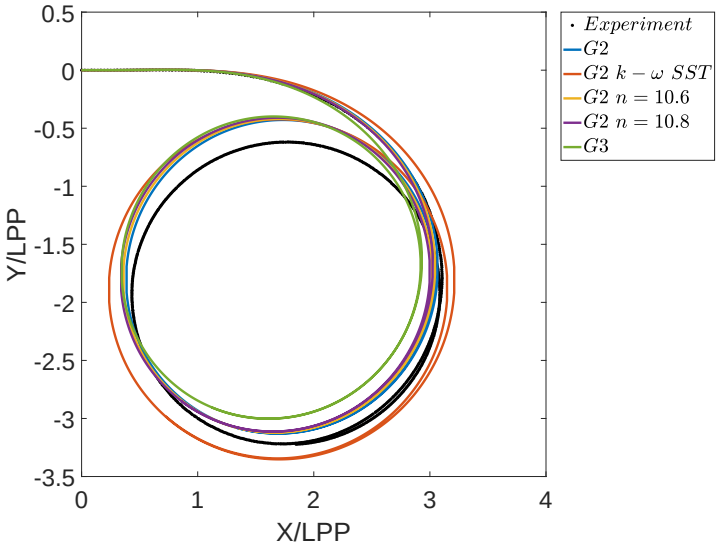

Figure 4.10: KCS turning circle maneuver trajectories with different data-driven propeller and rudder models using the G1 $\Delta z/2$ VOF grid with the baseline time step.

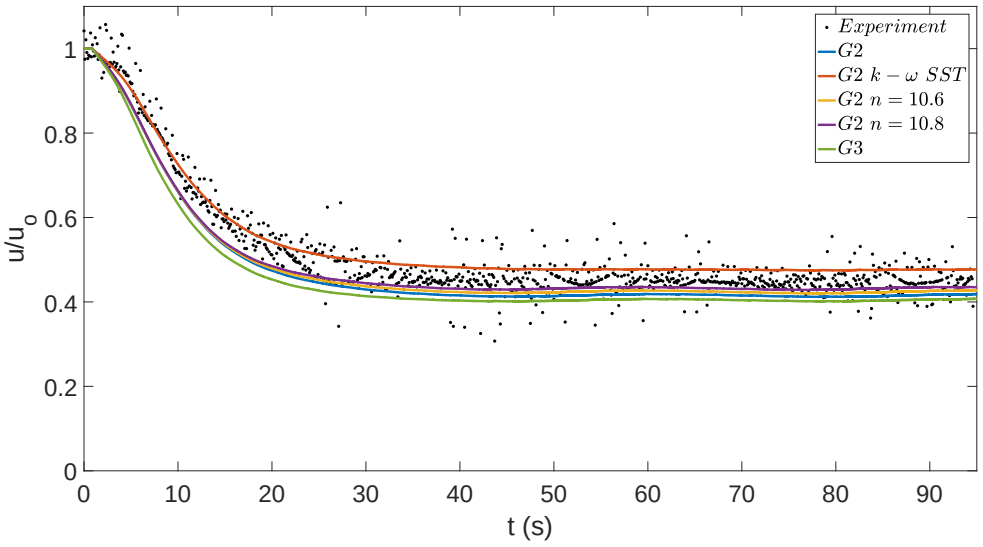

Figure 4.11: KCS turning circle maneuver dimensionless forward velocity with different data-driven propeller and rudder models using the G1 $\Delta z/2$ VOF grid with the baseline time step.

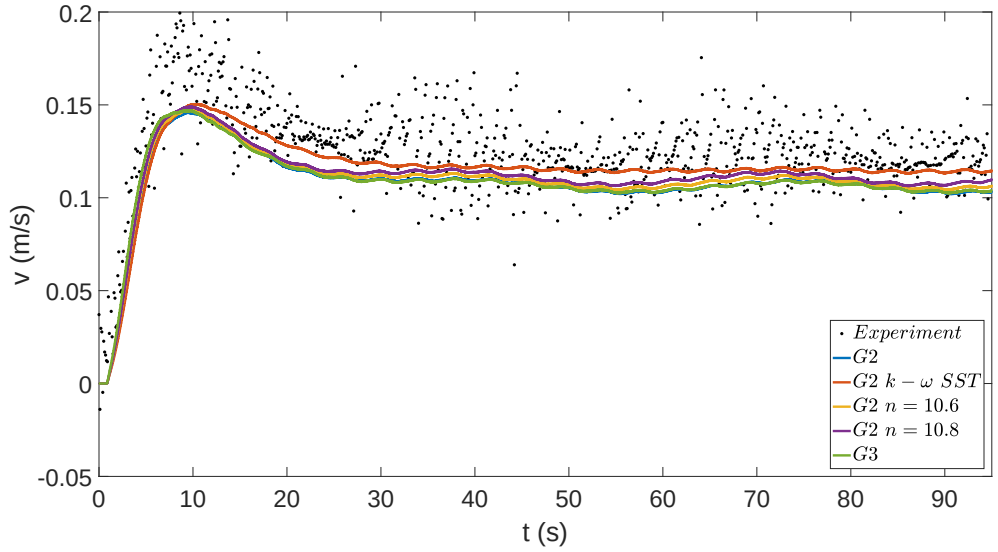

Figure 4.12: KCS turning circle maneuver sway velocity with different VOF data-driven propeller and rudder models using the G1 $\Delta z/2$ VOF grid with the baseline time step.

The `G2` $k-\omega$ `SST` data-driven model produces the largest turning circle, which over-predicts the experimental maximum advance, maximum transfer, and steady turning diameter by 3.2%, 3.4%, and 2.9% respectively. The sway velocity for this case is in good agreement with the experiment. The forward velocity predicted with this model is larger than the other CFD cases and the experiment. Furthermore, this model was shown to under-predict the rudder force, thus, since the rudder force is under-predicted and the forward velocity is over-predicted the turn is wider than the experiment. On the other hand, the cases that use the Spalart Allmaras turbulence model for training lead to a tighter turn with the maximum advance under-predicted by 1.0%, 1.8%, 2.4%, and 5.5% for the `G2`, `G2` n`=10.6 rps`, `G2` n`=10.8 rps`, and `G3` cases respectively. In the same order the maximum transfer is under predicted by 2.8%, 3.1%, 3.3%, and 6.8%; while the turning circle diameter is over-predicted by 2.7%, 3.8%, 3.9%, and 0% respectively. One thing to note is that when the propeller is run at a faster rate (n increasing) the turn becomes tighter since the load on the

rudder also goes up due to the induced velocity. Of course, when n is increased, the vessel forward velocity also increases as shown by Fig. 4.11. The `G3` case most significantly under-predicts the forward velocity of the experiment, while as noted the `G2` $k-\omega$ `SST` model has the largest steady forward velocity. The difference between the steady forward velocity between these two cases is nearly 15%. There is also a similar variation in terms of the sway velocity.

This illustrates how a data-driven propeller model can be used to evaluate the maneuvering characteristics of a vessel in calm water, but, the uncertainty of the data used to train the models propagates into the data-driven model. Depending upon which set of data is used to train the model, the maneuvering trajectory can either be tighter or wider than the experimental results, and the forward velocity can either be greater or less than the experiment. This illustrates that if high quality training data can be generated a very accurate data-driven model can also be constructed.

4.2.4 Computational Cost Discussion

The computational cost of both the training simulations for the data-driven model as well as the maneuvering simulations are important to consider. The methods discussed in this paper allow for a significant reduction in computational cost compared to using a discretized propeller. When the baseline time-step Δt_o is used, less than 2000 core hours are required to perform the turning circle simulations, which includes the twelve second period of time during which the vessel is accelerated from rest to before the rudder is actuated, as well as the nearly 100 s of the model-scale simulation time after the rudder is actuated.

In comparison, the overset grid analysis performed by *Shen et al.* (2015) showed that a larger KCS model with λ_s of 1/52.667 and a F_n of 0.26 required over 11,000 core hours to model 26 s of model time for the vessel performing a zig-zag maneuver with rotating discretized propellers and rudders on a grid with nearly four million cells.

Adwaith (2020) examined the KCS, with λ_s of 1/37.9 and a F_n of 0.26, performing a turning circle with different approaches for modeling the propeller including the use of a discretized propeller with the overset method. This work showed that for a 6.97 million cell grid, only three quarters of a turn cost over 100,000 core hours when the propeller was discretized. This study also showed that propeller models can reduce the computational cost.

When a propeller model is used it substantially increases computational efficiency. As noted earlier in the discussion on the temporal discretization, the time-step can be increased further to make the computational cost even lower. The time required to perform the VOF maneuvering simulations is independent of the training cost for the data-driven propeller model; however, depending upon the spatial fidelity and quantity of training data used to generate the data-driven propeller and rudder model, the training cost is variable. A substantial benefit of the methods used here is that the VOF maneuvering analysis can be performed on a grid with much lower spatial discretization than the CFD simulation with a discretized rotating propeller, and the temporal discretization can be much larger when performing the maneuver.

The training costs associated with the double-body grid CFD training simulations are: 800 core hours for the G1 grid, 3,000 core hours for the G2 grid, and over 34,000 core hours for the G3 grid. The sample point correlating to LHS point two is used to generate the reference training cost. To note, all cases have a time step limit such that the propeller rotation does not exceed 1° per time-step and also have a Courant number limit which drives up the computational cost for the G3 grid since the time step taken is less than 1° per time-step. Also, 40 cores are used for the G1 grid, 120 cores are used for the G2 grid, and 200 cores are used for the G3 grid. Some parallelization effects drive up the computational cost for the G3 grid since the efficiency of each core hour is less when many cores are used. Table 4.4 shows the length of time (in model scale) that the discretized propeller is modeled as a function

Table 4.4: Cost of double-body CFD simulations for KCS.

	1 point	4 points	8 points	16 points	28 points
Model time (s)	5.7	22.8	45.6	91.2	159.6
G1 (core hours)	800	3200	6400	12800	22400
G2 (core hours)	3000	12000	24000	48000	84000
G3 (core hours)	34000	136000	272000	544000	952000

of the number of sample points used to train the data-driven model. The table also shows the number of core hours that are necessary to perform one, four, eight, sixteen, and twenty-eight double-body CFD simulations.

It is also useful to view the computational cost graphically in terms of the cost of the implemented model in comparison to the cost if the propeller is discretized. Fig. 4.13 shows the cost of a double-body CFD simulation on the G2 grid with a discretized rotating propeller with the solid blue line, the implemented cost of the data-driven model trained with the eight point set on the G2 double-body grid with the dashed blue line, and the cost to train the data-driven model with the solid green line. This highlights that the computational cost of the implemented model is dominated by the training cost, and that the cost to model a maneuvering ship is significantly less expensive than modeling a discretized propeller. The solid blue line shows the cost of modeling a discretizized propeller operating in the behind condition, but with a double-body approximation, which is less expensive than the cost of using a discretized propeller for a VOF maneuvering analysis. Thus, this plot shows a lower bound for the comparison between the model computational cost and the discretized propeller computational cost for a maneuvering vessel; in reality, the computational savings using the model are even more substantial. The blue dot highlights the cross-over point at which point it is less expensive to train the data-driven model and use it to model the maneuvering vessel with the G1 $\Delta z/2$ VOF grid. The time to complete one circle is 61.2 s in model time after the rudder is actuated. Thus, it is less expensive to train and implement the model than use a discretized propeller even

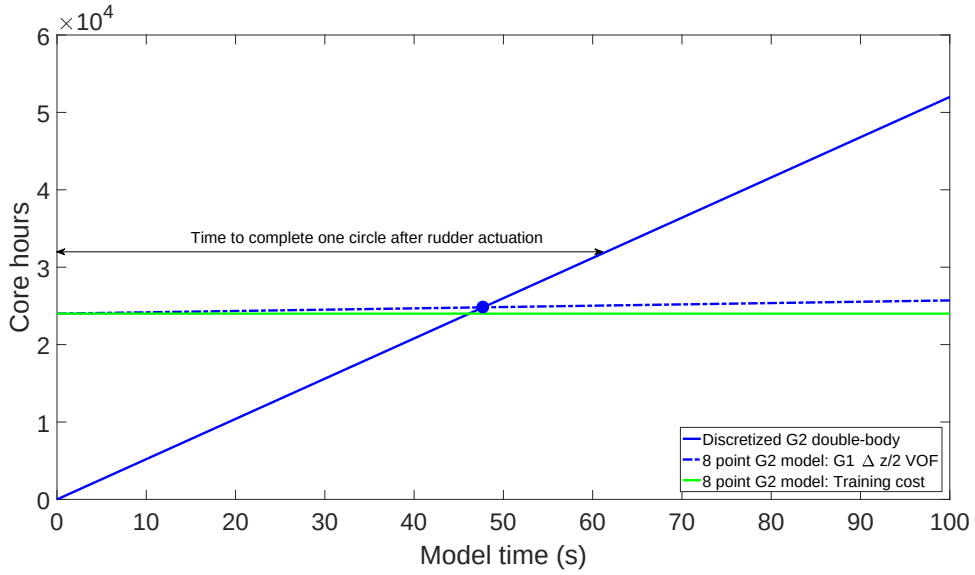

Figure 4.13: Computational cost to train and implement a data-driven model trained with eight points on the G2 double-body grid and implemented on the G1 $\Delta z/2$ VOF grid compared to using a discretized propeller on the G2 double-body grid.

if only one turning circle is examined, but this approach is especially applicable if multiple maneuvers are examined.

The left hand side of Fig. 4.14 shows the training cost for the G1, G2, and G3 models with different sets of training data, for which the sample point correlating to LHS point two is used to generate the reference training cost. The right hand side of Fig. 4.14 shows the cost of training a model using eight samples of double-body CFD simulation on the G1, G2, and G3 grids with a discretized rotating propeller. For each color the solid line represents the cost using a double-body discretized propeller simulation while the implemented cost of the data-driven model trained with the eight point set for each respective double-body grid is shown with the dashed lines. This highlights that the computational cost of the implemented model is dominated by the training cost for all grids examined, and that the cost to model a maneuvering ship is significantly less expensive than modeling a discretized propeller. The dots show the cross-over point at which point it is less expensive to train the data-driven model

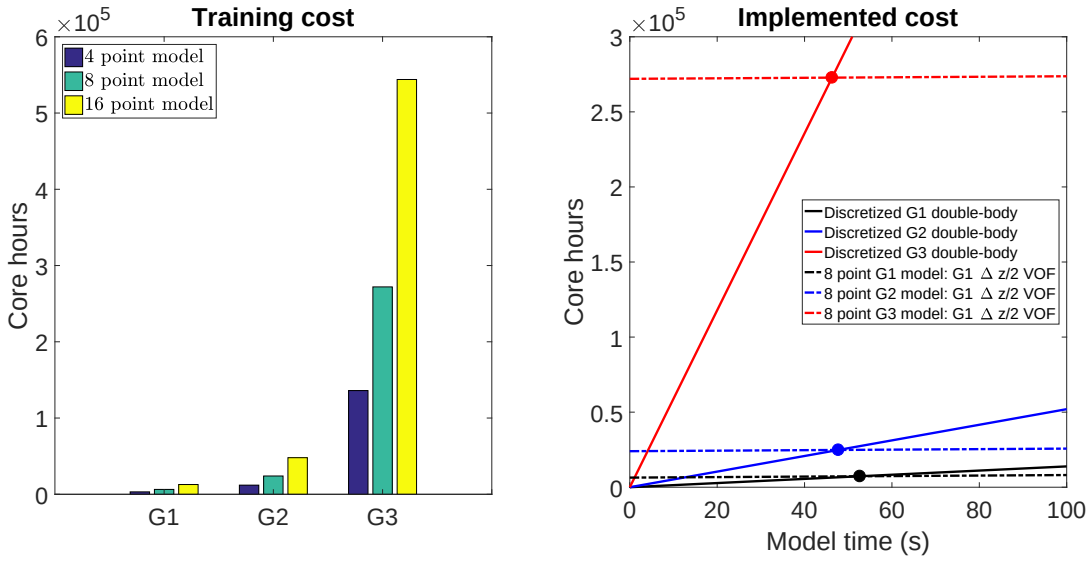

Figure 4.14: Left: Computational cost to train different sets of training data. Right: Computational cost to train and implement a data-driven model trained with eight points on the different double-body grids and implemented on the G1 $\Delta z/2$ VOF grid compared to using a discretized propeller.

and use it to model the maneuvering vessel with the G1 $\Delta z/2$ VOF grid. The time to complete one circle is 61.2 s in model time after the rudder is actuated. Thus, it is less expensive to train and implement the model for all of the different cases than use a discretized propeller even if only one turning circle is examined. Furthermore, it is illustrated that as finer training grids are used, the cross-over point moves to the left, thus as finer grids are used to train the model, they can be implemented on a coarser VOF grid for maneuvering simulations which reduces cost.

Figs. 4.13-4.14 highlight a distinct advantage of this method which is that a high fidelity simulation with small temporal and spatial discretization on the propeller and rudder can be used to train the model (G2 or G3), while the VOF maneuvering simulation can be performed on a much coarser grid (G1). Table 4.4 illustrates that if eight samples are used, the double-body training time is less than 50% of the time that it would take to perform the turning circle maneuver. It has been shown that generating a linear regression based data-driven propeller and rudder model with eight samples

on the G2 grid leads to good results with regards to the experiment. The VOF maneuvering simulations performed on the G1 grid with extra refinement around the waterplane costs less than 2000 core hours, and the training cost is approximately 24000 core hours. This illustrates, that even if just one turning circle maneuver is performed, the approach used in this work reduces the computational cost of performing a single maneuver (which requires over 100 s of model time simulation) by over 50% when both training cost and implementation modeling cost are accounted for. Furthermore, if multiple simulations are performed, the computational cost of the maneuver is small relative to the training cost. Thus, if it is desired to perform multiple maneuvering simulations, this method is especially appropriate. This method enables the computationally efficient analysis of multiple Froude numbers, the effect of different hull displacements, the effect of small hull geometry modifications, and the analysis of the vessel operating in waves.

4.3 Turning Circle in Waves

Similar to the case of calm water, the data-driven propeller model is compared to the experimental results. First, the propeller and rudder forces are evaluated in relation to the experimental forces using the experimental motions. Second, the motions using the data-driven propeller and rudder model are compared to the experimental motions. For the maneuver in waves, the experimental propeller revolution rate is increased to n equal to 13.2 rps and the initial forward speed of the vessel at the time the rudder is actuated is 0.86 m/s. The CFD model uses the same initial forward speed and the propeller model uses the experimental value of n.

4.3.1 Comparison of Model Forces with Experimental Forces in Waves

Several linear regression based data-driven models are evaluated by using the experimental motions of the vessel as inputs to the models and comparing the output

propeller thrust, propeller torque, and rudder normal force of the movable part of the rudder to the experimental force calculations. All of the cases except for one are trained with CFD data using the Spalart Allmaras turbulence model. The one exception is the `8 point G2` $k-\omega$ `SST` case which is trained with the eight point LHS set results on the G2 grid using the $k-\omega$ SST turbulence model.

The nomenclature for naming the models is the same as earlier. In the name of the model, the first number denotes the number of training points used, and the number following the `G` specifies which grid was used to develop the training data. The eight point and sixteen point LHS sets on both the G1 and G2 double-body grids are used to train the `8 point G1` model, the `16 point G1` model, `8 point G2` model, and the `16 point G2` model. Finally, the eight point LHS training set on the G3 grid is used to develop the `8 point G3` model.

Fig. 4.15, Fig. 4.16, and Fig. 4.17 show the thrust, torque, and rudder normal force respectively for the different linear regression based models as well as the experimental predictions. The rudder is actuated at 0 s, but the flow has been established over the hull before this point. The propeller and rudder models neglect the rudder actuation and assume that the rudder is fully deflected. Again, it is important to note that the force predictions made here use the experimental motions as input. It can be seen that the predicted thrust and torque are slightly less than the mean experimental values, but also within the bounds of the variation in thrust as a function of time. The model trained with CFD using the $k-\omega$ SST turbulence model predicts the largest thrust but also predicts the least rudder normal force, as was the case in the calm water analysis. The rudder models trained with the eight point LHS set of data on either the G2 grid or the G3 grid predict larger rudder normal force than the other models. The G3 grid case predicts the largest rudder normal force. That said, the average rudder normal force predicted by the linear regression based models under-predicts the average experimental force.

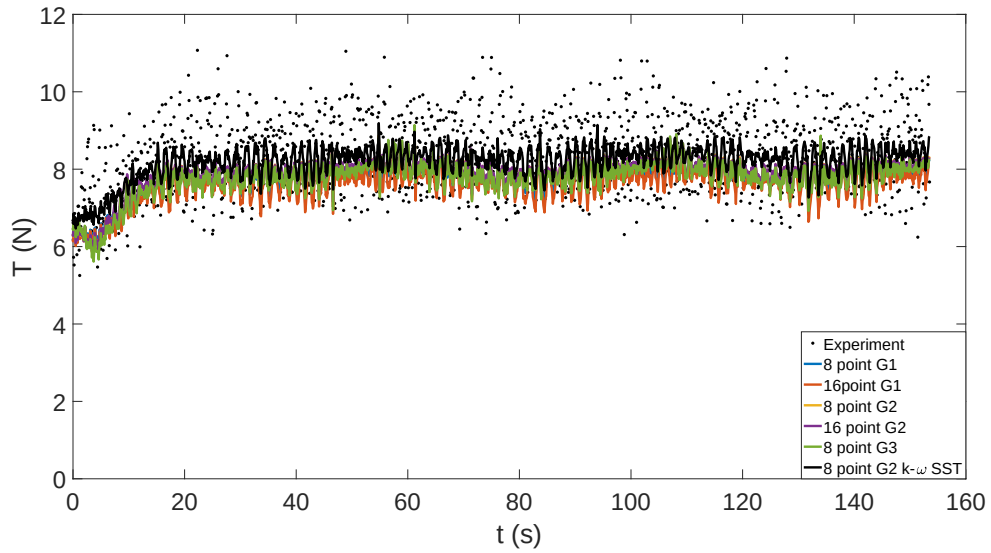

Figure 4.15: Experimental thrust compared to data-driven thrust using motions of experiment in waves.

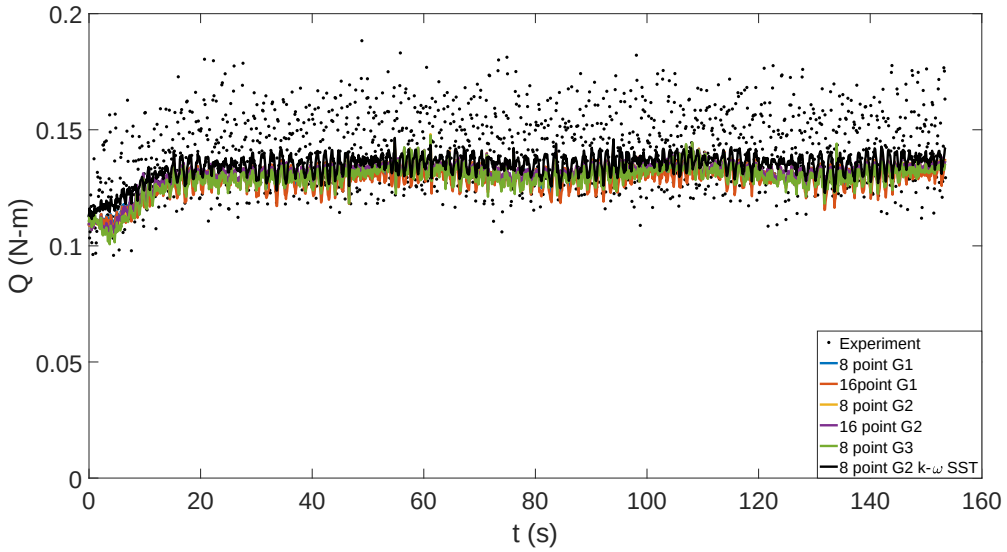

Figure 4.16: Experimental torque compared to data-driven torque using motions of experiment in waves.

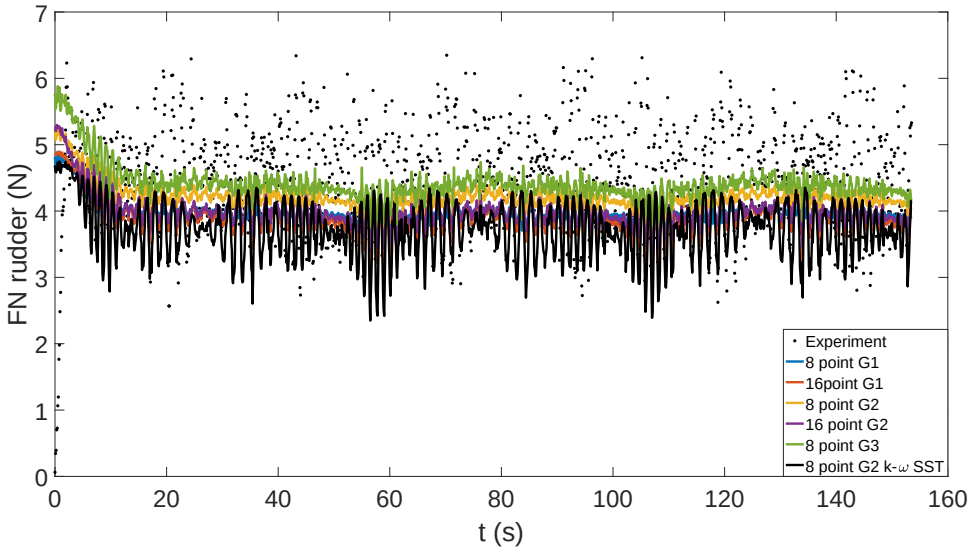

Figure 4.17: Experimental rudder normal force compared to data-driven rudder normal force using motions of experiment in waves.

To further investigate the discrepancies between the different CFD models and the experiment, the forces are analyzed during the steady part of the turn (after 40 s of model time after the actuation of the rudder). Table 4.5, Table 4.6, and Table 4.7 show the tabulated predictions of the thrust, torque, and rudder normal force as well as the error of the model with respect to the experiment ϵ_m. These tables illustrate that the propeller and rudder trained with the $k-\omega$ SST turbulence model results agree best with the experiment for thrust and torque, but the worst for the prediction of the rudder normal force. The models trained with the G1 grid contain too much discretization error and do not agree well with the experiments. The difference between the use of the eight point versus the sixteen point LHS set to train the model on the G2 Grid leads to small differences in ϵ_m for the thrust and torque, but the sixteen point training set leads to worse prediction of the rudder normal force. For these reasons, the most interesting cases to examine in more detail are the cases trained with the eight point LHS set on the G2 with differences in turbulence modeling approach and the effect of using the finer G3 grid.

Table 4.5: Mean error using data-driven model thrust predicted with experimental motions for KCS in waves after 40 s.

Model	T (N)	ϵ_m (%)
Experiment	8.461	0.000
8 point LHS G1	7.852	7.198
16 point LHS G1	7.736	8.570
8 point LHS G2	7.963	5.895
16 point LHS G2	8.009	5.341
8 point LHS G3	7.907	6.550
8 point LHS G2 $k-\omega$ SST	8.299	1.923

Table 4.6: Mean error using data-driven model torque predicted with experimental motions for KCS in waves after 40 s.

Model	Q (N-m)	ϵ_m (%)
Experiment	0.143	0.000
8 point LHS G1	0.131	8.246
16 point LHS G1	0.130	9.281
8 point LHS G2	0.132	7.396
16 point LHS G2	0.133	7.195
8 point LHS G3	0.131	8.453
8 point LHS G2 $k-\omega$ SST	0.136	5.088

Table 4.7: Mean error using data-driven model normal force predicted with experimental motions for KCS in waves after 40 s.

Model	FN (N)	ϵ_m (%)
Experiment	4.441	0.000
8 point LHS G1	3.948	11.098
16 point LHS G1	3.893	12.340
8 point LHS G2	4.168	6.158
16 point LHS G2	3.949	11.071
8 point LHS G3	4.334	2.408

4.3.2 Vessel Motions with CFD Versus Experiment in Waves

Several data-driven propeller and rudder models are applied to a maneuvering CFD analysis of the vessel operating in regular waves. The vessel is in head seas with a height of 48 mm, a wave length equal to the ship length, with a wave period of 1.4 s. The square G1 $\Delta z/2$ VOF grid is used for all cases. The propeller revolution rate applied to the propeller model is 13.2 rps, which is the same as the experiment. The initial forward speed of the vessel is 0.86 m/s.

As noted earlier when the experimental propeller and rudder forces were compared to different data-driven model predictions for the vessel operating in waves, there is some discrepancy between different models based on the data with which they are trained. To highlight the ways that these differences manifest themselves in a maneuver in waves four different linear regression based propeller and rudder models are used in a VOF CFD analysis. The training sets include several variations that use the eight point LHS set of data. The G2 set of training data that is generated with the Spalart Allmaras turbulence model (denoted `G2`), the G2 set of training data with the $k-\omega$ SST turbulence model (denoted `G2` $k-\omega$ `SST`), and the G3 set of training data trained with the Spalart Allmaras turbulence model (denoted `G3`).

Fig. 4.18 shows the trajectory of the vessel predicted with the different CFD approaches and the experiments, for which the rudder is deflected at the origin. Similar to the calm water turning circle results, the `G2` $k-\omega$ `SST` model over-predicts the experimental maximum advance and maximum transfer. When a vessel operates in waves, a second order hydrodynamic force acts on the vessel which causes it to drift. Overall, the second order drift angle of the CFD results is similar to the experiments, but the drift distance is substantially less. The G2 $k-\omega$ SST case agrees best with the experiment in terms of second order drift distance. Overall, the G3 grid predicts a tighter circle with a similar drift angle and distance as the G2 grid.

Fig. 4.19 and Fig. 4.20 show the dimensionless forward velocity and sway velocity

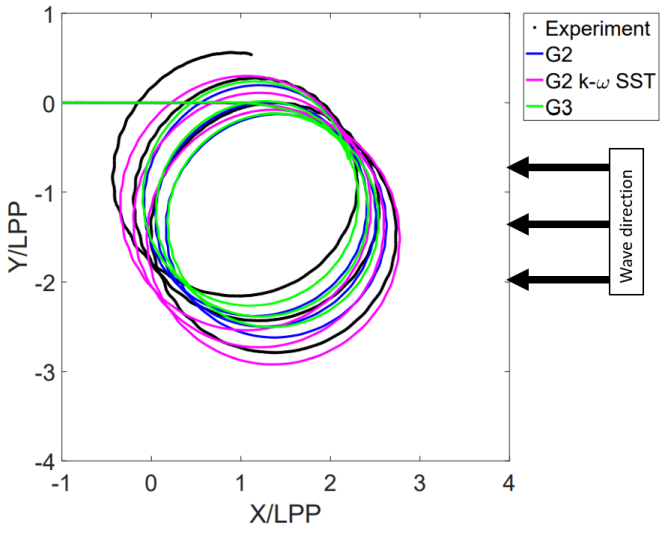

Figure 4.18: Trajectory of for KCS turning circle maneuver in waves with different variations of the linear regression based propeller and rudder model on the square G1 $\Delta z/2$ VOF grid.

as a function of time during the maneuver. Good agreement between the `G2` $k-\omega$ `SST` model and the experiment is illustrated for the forward speed. This is again sensible since when the forces were calculated using experimental motions, this model predicted larger thrust and less rudder force than the other models, which leads to a wider turn, but a higher forward velocity. The `G3` based model produces slightly lower forward velocity predictions compared to the G2 grid based models. All of the sway velocities predicted with the CFD are similar and are near the experimental mean, but the amplitude of the oscillation predicted with the experiment is larger than that predicted with the CFD.

Differences in the way that the propeller and rudder model are trained, particularly the fidelity of the training data, can manifest in noticeable differences in maneuvering characteristics. This illustrates that as the uncertainty in the underlying CFD training data decreases, the uncertainty in the maneuvering prediction also improves. The most substantial difference is found with regards to the effect of using different turbulence models to train the methods.

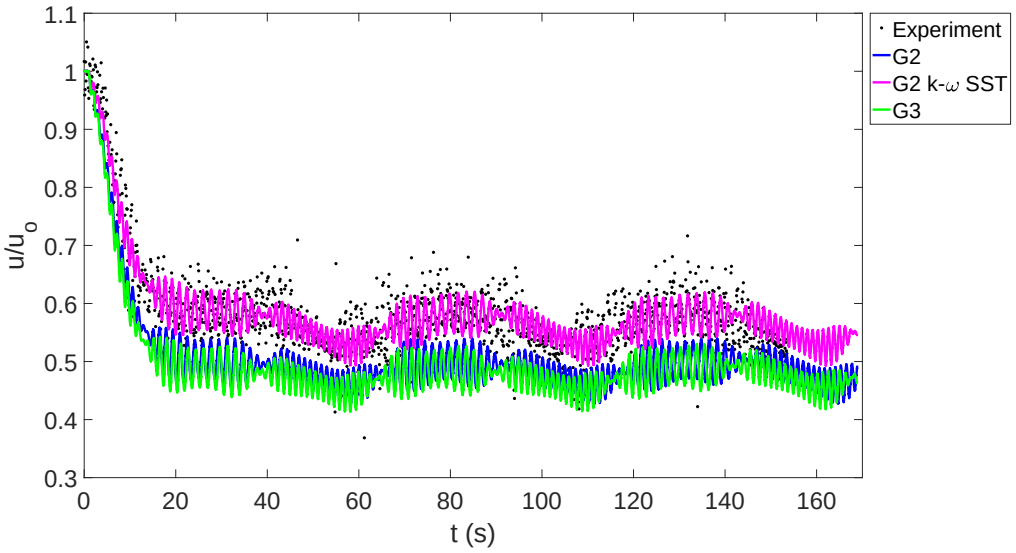

Figure 4.19: Forward velocity for KCS turning circle maneuver in waves with different variations of the linear regression based propeller and rudder model on the square G1 $\Delta z/2$ VOF grid.

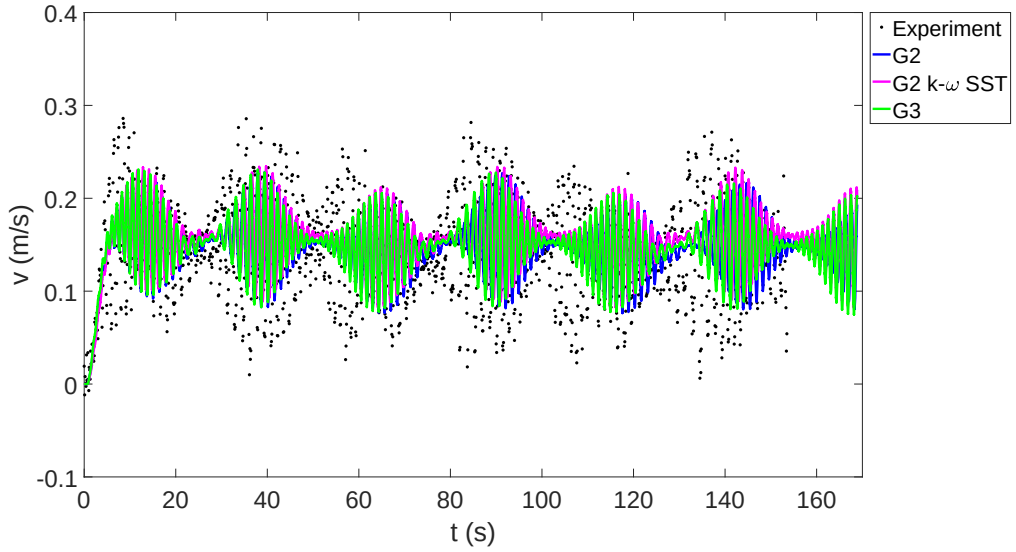

Figure 4.20: Sway velocity for KCS turning circle maneuver in waves with different variations of the linear regression based propeller and rudder model on the square G1 $\Delta z/2$ VOF grid.

CHAPTER V

Data-Driven Model Development of the Propellers and Rudders for the ONR Tumblehome

This chapter develops different data-driven models for the propellers and rudders of the ONR Tumblehome for application to turning circle maneuvers both in calm water and in waves. A select group of data-driven models for the propellers and rudders are applied to CFD maneuvering simulations of the vessel in Chapter VI. Table 5.1 shows the parameters of the vessel.

In this chapter the modeling techniques are discussed. Double-body RANS CFD with a discretized rudder and rotating propeller are used to generate training data for the models. The double-body RANS CFD training cases are discussed first. Following the discussion on the CFD setup, the data-driven model generation is discussed. A series of data-driven models are trained with the results of the training CFD. The data-driven models are used to predict the K_X, K_S, C_{Fx}, C_{Fy}, and C_{Mz}. Two methods are used to generate the data-driven models: linear regression and Gaussian process regression. The discrepancies between different regression based models are discussed in terms of discretization error, turbulence modeling error, and error in the data-driven model itself. The asymmetry between the port and starboard propellers and rudders is also highlighted.

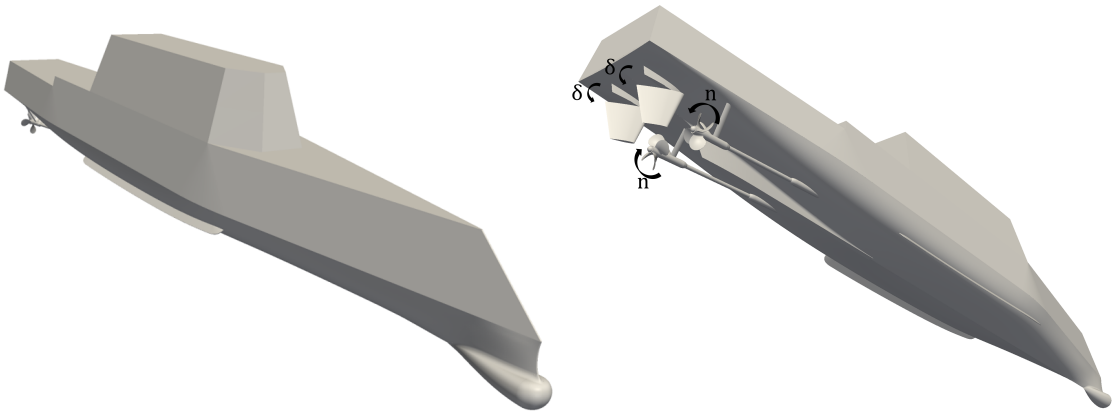

Figure 5.1: ONR Tumblehome geometry. Left: Vessel viewed front. Right: Vessel viewed from underneath with propeller inward rotation direction and rudder angles shown.

Table 5.1: Parameters for ONR Tumblehome geometry and case setup.

Parameter	Value
Length of Waterline (L_{WL})	3.147 m
Beam of Waterline (B_{WL})	0.3838 m
Draft	0.1435 m
Model Scale (λ_s)	1/48.935
Displacement	72.6 kg
Initial velocity u_o	1.11 m/s
Rudder angle δ	-35 ^{o}
Lateral area of rudder S_R	0.012 m^2
Axial distance from center of gravity to rudder stock d_r	1.384 m
Number of propeller blades	4
Propeller diameter (D)	0.1066 m
Propeller shaft angle	5°
Axial distance from center of gravity to propeller d	1.291 m
Lateral distance from centerline to propeller d_y	0.084 m

5.1 CFD Model Development

OpenFOAM version 2.4.x is used for the analysis. The grids are generated using the OpenFOAM utilities blockMesh and snappyHexMesh. Like in the analysis of the KCS, two separate CFD models are created. The simulation used for training the data-driven model uses a double-body approximation at the waterplane and models each of the propellers rotating. For the maneuvering CFD simulations the VOF approach is used to capture the interface between the water and air and the data-driven model is used to determine the force of both propellers and both rudders.

5.1.1 Double-Body CFD of Discretized Propeller and Rudder

The double-body RANS CFD of the hull, discretized propellers and twin rudders is run using the OpenFOAM solver pimpleDyMFoam. Both rudders are deflected to a δ of -35 degrees. The propeller is rotated at no more than one degree per time-step using a rotating sliding mesh. Three grids are examined, the G1 grid has 2.5 million cells, the G2 grid has 6.2 million cells, and the G3 grid has 16.4 million cells. Fig. 5.2 shows a slice of each grid around the propellers and rudders. The propeller rotates at an n of 8.97 rps and the duration of each simulation is 5.57 s which correlates to 50 propeller revolutions. The data-driven model uses the average force on each propeller and each rudder for the second half of the simulation time for training and validation. This allows the flow to develop over the vessel, such that steady state forces are used to train the data-driven model.

The CFD domain is 4 L_{WL} long, 3 L_{WL} wide, and 1 L_{WL} deep. The vessel is located at the top of the domain in the longitudinal and lateral center of the domain. The top of the domain is a plane of symmetry. The bottom of the domain, the side at maximum value of negative Y, and the side at the maximum positive X are velocity inlets. The sides of the domain at the maximum value of positive Y and negative X are outlets. The hull, propeller, rudder are non-slip walls modeled with wall functions.

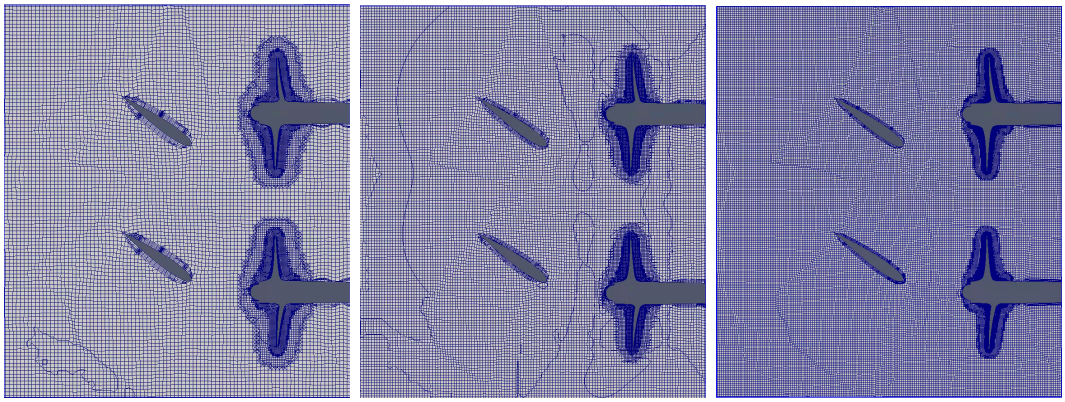

Figure 5.2: Slice of ONR Tumblehome double-body grid at the 0.11 m below the waterplane with Z normal. Left: G1 grid. Middle: G2 grid. Right: G3 grid

The bow is pointed in positive X, positive Y is towards port, positive Z is vertical upwards. Figure 5.3 shows the CFD domain on the left as well as the surface grid of the propellers and rudders on the right. In the image of the CFD domain the vessel is shown in red and the mesh on the boundaries is also shown.

In the region around the hull, extending +/- 1.9 m fore and aft from the vessel center of gravity, 0.24 m laterally in each direction, and from the waterplane to 0.14 m below the waterplane the cells of the grid are nearly isotropic, such that the cell discretization length in the X, Y, and Z directions are within 3.5% of the characteristic length of each cell Δ_g. Δ_g is the average of the cell length in X, Y, and Z directions. Stretching is applied to the cells in the regions outside of this region to reduce grid count as illustrated in Fig. 3.3. SnappyHexMesh is used to refine the grid, for which each level of refinement halves the Δ_g. Different levels of refinement are applied to each surface. One level of refinement is applied to the hull. Two levels of refinement are applied to the `Misc. Appendages`: the bulb, bilge keels, struts, and propeller shaft. Three levels of refinement are applied to the rudder and five levels of refinement are applied to the propeller. Five prism layers with a thickness of 0.3 of the local Δ_g are applied to each surface. Three levels of refinement are applied

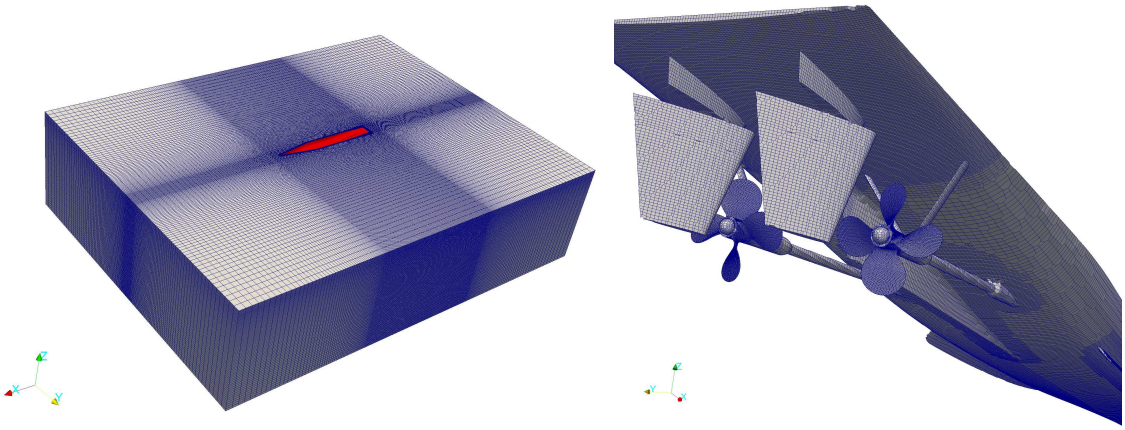

Figure 5.3: ONR Tumblehome Double-body G2 mesh. Left: Domain for double-body simulations of ONR Tumblehome. Right: Surface grid on ONR Tumblehome propeller and rudder.

Table 5.2: ONR Tumblehome double-body CFD grid details.

Grid	G1	G2	G3
Isotropic region $\Delta_g(m)$	2.76E-02	1.95E-02	1.38E-02
Refinement zone Δ_g (m)	3.45E-03	2.44E-03	1.73E-03
Hull Δ_g (m)	1.38E-02	9.76E-03	6.90E-03
Misc. Appendages Δ_g (m)	6.90E-03	4.88E-03	3.45E-03
Rudder Δ_g (m)	3.45E-03	2.44E-03	1.73E-03
Propeller Δ_g (m)	8.63E-04	6.10E-04	4.31E-04

in the region around the propellers and rudders to better capture the complex flow. Table 5.2 shows the mesh size on each grid for the different regions and surfaces.

5.2 Data-Driven Model Development for ONR Tumblehome

The data-driven model is generated in a similar way as it was for the KCS. The main exceptions to this are that for the ONR Tumblehome the model must be generated for each propeller-rudder pair and a different operating J and β space is examined. Three distinct sets of training/validation data are generated corresponding to a four point LHS set, an eight point LHS set, and sixteen point LHS set. These

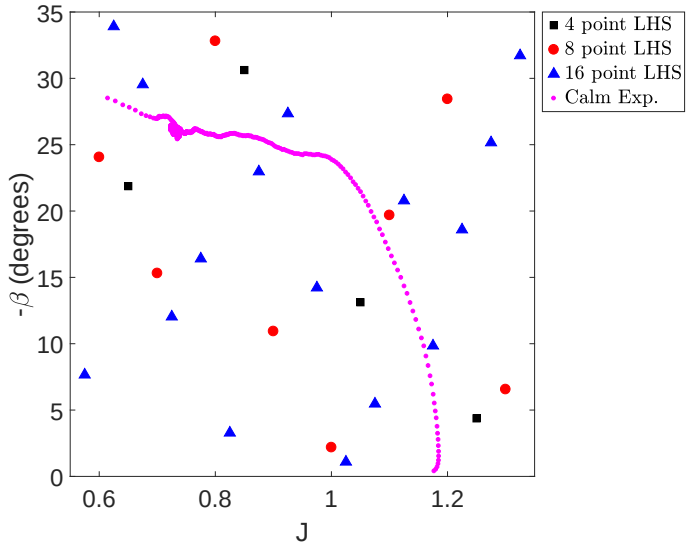

Figure 5.4: LHS points for training the ONR Tumblehome data-driven model with experimental data for F_n=0.2, δ=-35° turning circle (*SIMMAN* (2020)).

samples are examined in the J-β space correlating to a turning circle maneuver for a Froude number of 0.2 with the rudder deflected to -35°. For this maneuver the starting forward velocity is 1.11 m/s. Fig. 5.4 show the sample points along with the J and β values for the calm water turning circle in the *SIMMAN* (2020) experimental results. The tabulated values of J, β, u, and v, are shown in Table B.1. LHS points 1-4 correlate to the four point LHS set, LHS points 5-12 correlate to the eight point LHS set, and LHS points 13-28 correlate to the sixteen point LHS set.

The data-driven model is generated using the regression based data-driven model. The four point LHS set is used to validate the model, while the eight point and sixteen point sets are used to train the model. The models are also trained with both the Spalart Allmaras turbulence model and the $k-\omega$ SST turbulence model so that the differences between turbulence models can be characterized. Additionally, the effect of grid density is examined. Just like in the KCS set of data, there is error and uncertainty in the data-driven model. Discretization error and turbulence modeling error is inherent in the training and validation data. Additionally, there is error in

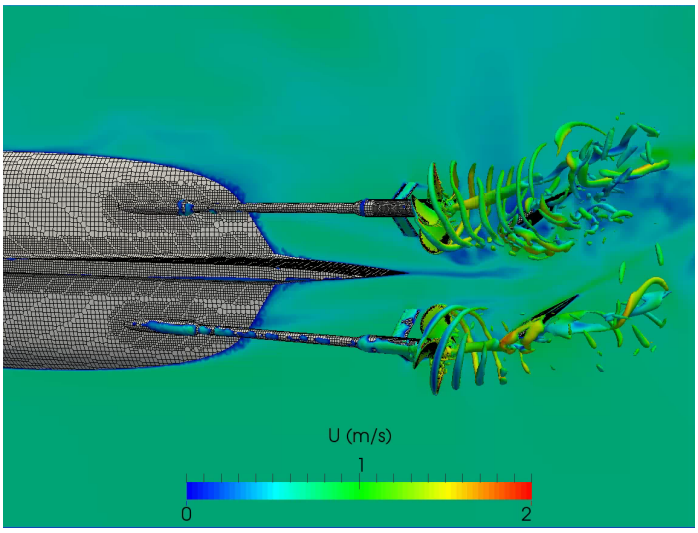

Figure 5.5: Flow velocity over the propeller and rudder with Q criterion of 2500 shown along with a slice across the rudder plane illustrating asymmetric flow and separation on the LHS point seven data-driven model training case with 16.4 million cells.

the model itself that is used to predict the propeller and rudder forces.

The forces on the propellers and rudders are asymmetric. Fig. 5.5 illustrates the asymmetry of the flow for LHS point seven with the Spalart Allmaras turbulence model. A slice through the rudder plane colored by velocity magnitude is shown along with a contour of Q criterion equal to 2500, which illustrates the evolution of the tip vortex of the propeller blades washing over the rudder. The mesh on the surface of the hull, propeller shafts, propeller, and rudder are shown. The force of the propellers is asymmetric and the interaction of each propeller with each rudder is different. This image shows the view from the bottom of the vessel with the rudder deflected to -35° for this training case at a J of 0.6 at a $|\beta|$ of approximately 26°. With the rudder deflected to this angle the vessel has a positive yaw rate and a negative sway velocity. Thus, the starboard side propeller and rudder see a clean inflow, while the port side propeller and rudder have a more perturbed inflow due to the wake of the hull. The starboard side rudder sees a lower angle of attack and produces less force than the port side rudder which sees a large angle of attack and has larger force.

The data-driven model is used to resolve the forces on each propeller and rudder across the operational J-β space. The error sources are discussed in terms of discretization error, turbulence modeling error, and the error in the model prediction. After the model is constructed and evaluated, the model is applied to the vessel performing turning circle maneuvers. The propeller force coefficient in the surge degree of freedom direction of the vessel K_X, the propeller side force coefficient K_S, the rudder resistance coefficient aligned with the surge degree of freedom direction C_{Fx}, the rudder side force coefficient C_{Fy}, and the rudder yaw moment coefficient C_{Mz} are analyzed. The K_X and the K_S are determined by Eqn. 2.13, and the rudder coefficients are determined by Eqn. 2.15-2.17. Each propeller and rudder force coefficient is calculated separately for the port and starboard sides. The forces of the propeller and rudder are calculated from the coefficients and the instantaneous motions of the vessel as described in Chapter II.

5.2.1 Discretization Error

The discretization error is important to quantify. The G1, G2, and G3 double-body grids with the discretized rudder and propeller are used to examine LHS points one through twelve. The propeller axial force coefficient K_X and the propeller side force coefficient K_S are considered. The rudder axial force coefficient C_{Fx}, the rudder side force coefficient C_{Fy}, and the rudder yaw moment coefficient C_{Mz} are also examined. The analysis of three grids with $\sqrt{2}$ refinement on the background mesh size shows oscillatory convergence for some of the cases examined.

Table 5.3 and Table 5.4 quantify the difference in the K_X prediction for the port side and starboard side propeller respectively for each of the grids and LHS points. These tables show the values for each LHS point, the prediction of K_X for each of the grids as well as the percent error with respect to the G3 value for the G2 grid and the G1 grid. The discretization error is calculated in the same way as Eqns. 3.2-3.3.

Table 5.3: Discretization error between different grids for ONR Tumblehome port side K_X.

LHS point	G1 (K_X)	G2 (K_X)	G3 (K_X)	ϵ_{23} (%)	ϵ_{13} (%)
1	0.313	0.321	0.330	2.653	5.137
2	0.386	0.386	0.381	-1.242	-1.166
3	0.298	0.310	0.316	1.731	5.651
4	0.252	0.259	0.256	-0.869	1.891
5	0.235	0.233	0.222	-4.647	-5.560
6	0.345	0.353	0.359	1.755	3.804
7	0.395	0.388	0.389	0.337	-1.594
8	0.335	0.338	0.349	3.344	4.005
9	0.181	0.194	0.197	1.644	8.112
10	0.359	0.368	0.381	3.555	5.747
11	0.415	0.434	0.435	0.163	4.672
12	0.216	0.217	0.216	-0.603	-0.189

The left hand side of Fig. 5.6 shows the average absolute value of the discretization error for K_X. The average of the absolute value of the percent error between the G3 and G2 grid ($\overline{|\epsilon_{23}|}$) for the K_X on the port side is 1.88% and 1.25% on the starboard side. The maximum absolute value of the percent difference between the G3 and G2 grids is less than 5%. On the other hand, the maximum absolute value of the percent difference between the G3 and G1 grids is over 8% on the port side LHS point nine case. On average the absolute value of the percent error between the G3 and G1 grid ($\overline{|\epsilon_{13}|}$) for K_X is larger, with 3.96% error on the port side and 1.8% on the starboard side. Since the port side propeller is on the leeward side it has less clean inflow compared to the starboard side, so it is sensible that the error between different grid densities is larger for the port side propeller.

The right hand side of Fig. 5.6 shows the average absolute value of the discretization error for K_S for each grid i, for G1, G2, and G3. The magnitude of the K_S is much smaller than that of the K_X, but is still important to consider in a maneuver. The K_S acts toward port when the rudder performs a maneuver for which v is negative (towards starboard) and/or the yaw rate is positive about the vertical Z

Table 5.4: Discretization error between different grids for ONR Tumblehome starboard side K_X.

LHS point	G1 (K_X)	G2 (K_X)	G3 (K_X)	ϵ_{23} (%)	ϵ_{13} (%)
1	0.445	0.441	0.443	0.583	-0.268
2	0.499	0.505	0.502	-0.603	0.636
3	0.332	0.332	0.331	-0.270	-0.468
4	0.255	0.254	0.247	-2.850	-3.198
5	0.322	0.320	0.306	-4.689	-5.243
6	0.336	0.336	0.332	-1.223	-1.227
7	0.523	0.530	0.526	-0.671	0.677
8	0.467	0.465	0.473	1.730	1.436
9	0.309	0.300	0.295	-1.494	-4.850
10	0.395	0.399	0.399	-0.107	0.827
11	0.470	0.478	0.477	-0.130	1.485
12	0.251	0.249	0.247	-0.681	-1.296

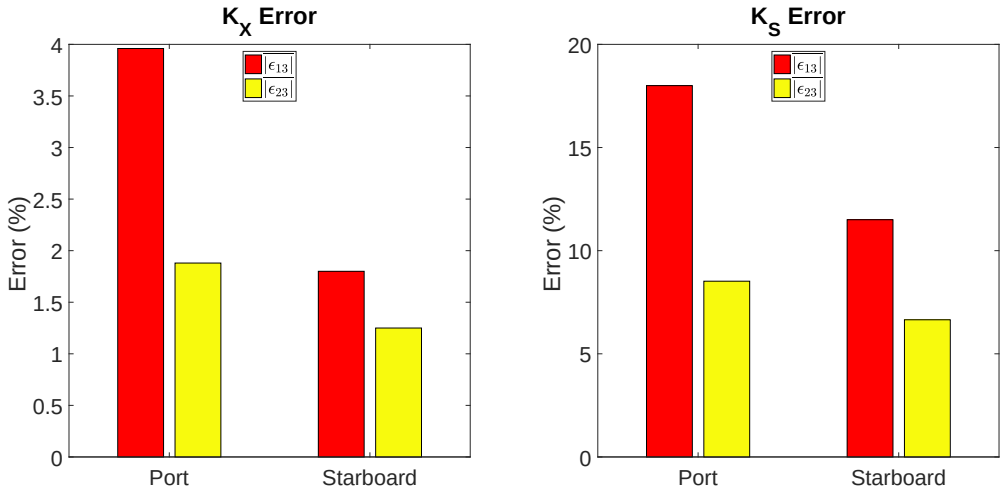

Figure 5.6: Average absolute value of the discretization error for K_X (left) and K_S (right) on port and starboard sides for G1, G2, and G3 for LHS points one through twelve.

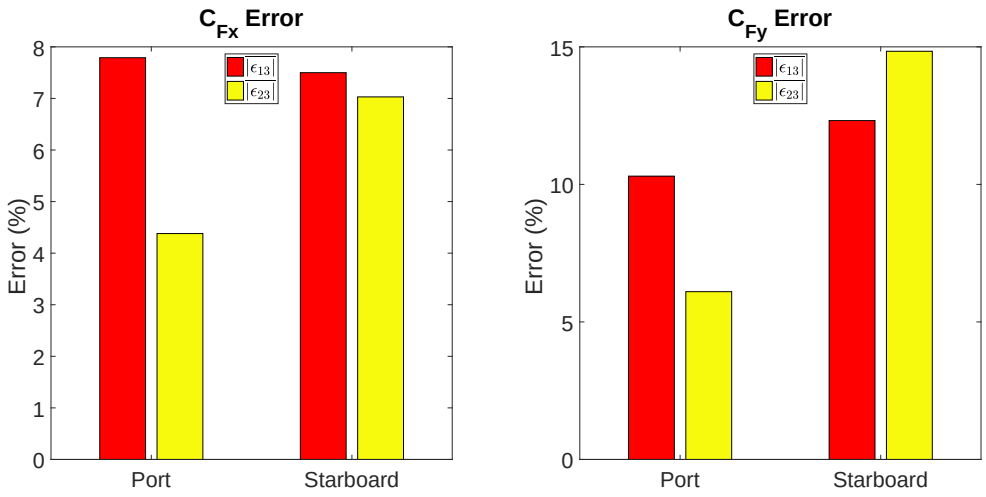

Figure 5.7: Average discretization error for C_{Fx} (left) and C_{Fy} (right) on port and starboard sides for G1, G2, and G3 for LHS points one through twelve.

axis. The propeller side force counteracts the force of the rudders. Additionally, when examined on the basis of percent error between different grids, the magnitude of the error tends to be larger than the K_X. Table B.2 and Table B.3 quantify the difference in the K_S prediction for the port side and starboard side propeller respectively for each of the grids and LHS points. On average, the port side values of K_S are 0.041, 0.043, and 0.041 for G1, G2, and G3 respectively. On average, the values of K_S on the starboard side are 0.067, 0.069, and 0.070 for G1, G2, and G3 respectively. Thus, the starboard side propeller (or the windward propeller) tends to produce a larger side force than the port side propeller. On the other hand, the K_X of the port side propeller tends to be larger as shown by Table 5.3 and Table 5.4.

Table B.4 and Table B.5 quantify the difference in the C_{Fx} prediction for the port side and starboard side rudder respectively for each of the grids and LHS points one through twelve. The left hand side of Fig. 5.7 shows the average absolute value of the discretization error for C_{Fx}; the right hand side of the figure shows the average absolute value of the discretization error for C_{Fy}.

Table 5.5 and Table 5.6 quantify the difference in the C_{Fy} prediction for the port

Table 5.5: Discretization error between different grids for ONR Tumblehome port side C_{Fy}.

LHS point	G1 (C_{Fy})	G2 (C_{Fy})	G3 (C_{Fy})	ϵ_{23} (%)	ϵ_{13} (%)
1	-0.333	-0.356	-0.369	3.338	9.553
2	-0.442	-0.442	-0.431	-2.527	-2.476
3	-0.330	-0.364	-0.359	-1.496	8.068
4	-0.394	-0.418	-0.484	13.768	18.738
5	-0.303	-0.325	-0.358	9.144	15.492
6	-0.386	-0.443	-0.482	8.061	19.936
7	-0.408	-0.392	-0.383	-2.279	-6.563
8	-0.351	-0.359	-0.370	3.109	5.057
9	-0.273	-0.282	-0.331	14.799	17.552
10	-0.389	-0.418	-0.411	-1.580	5.465
11	-0.429	-0.441	-0.423	-4.141	-1.385
12	-0.384	-0.403	-0.443	8.994	13.301

side and starboard side rudder respectively for each of the grids and LHS points one through twelve. On average the absolute value of the percent error between the G3 and G2 grid for the C_{Fy} on the port side is 6.10% and 14.84% on the starboard side. On average the absolute value of the percent error between the G3 and G1 grid for the C_{Fy} on the port side is 10.30% and 12.32% on the starboard side. Table B.6 and Table B.7 quantify the difference in the C_{Mz} prediction for the port side and starboard side rudder respectively for each of the grids and LHS points one through twelve. Since the yaw moment of the rudders are dominated by the side force of the rudders, with some influence from the resistance of the rudder, the discretization error is very similar to the results shown in Table 5.5 and Table 5.6.

Oscillatory convergence occurs for many of the the LHS points wherein the agreement between the G1 and G3 grids is better than the agreement between the G2 and G3 grids. Separation is present when a rudder operates at a high angle of attack, and this is difficult to resolve with RANS CFD with wall functions. These discretization errors are significant and influence the data-driven model when the model is constructed, but they would also influence the maneuvering simulation if a discretized

Table 5.6: Discretization error between different grids for ONR Tumblehome starboard side C_{Fy}.

LHS point	G1 (C_{Fy})	G2 (C_{Fy})	G3 (C_{Fy})	ϵ_{23} (%)	ϵ_{13} (%)
1	-0.143	-0.124	-0.143	13.401	0.092
2	-0.235	-0.236	-0.275	14.096	14.491
3	-0.497	-0.464	-0.526	11.813	5.523
4	-0.400	-0.429	-0.529	18.806	24.295
5	-0.383	-0.312	-0.449	30.433	14.799
6	-0.450	-0.453	-0.541	16.192	16.716
7	-0.202	-0.204	-0.234	12.597	13.448
8	-0.095	-0.081	-0.094	13.761	-0.603
9	-0.199	-0.219	-0.264	17.093	24.650
10	-0.485	-0.445	-0.500	10.909	2.959
11	-0.388	-0.312	-0.345	9.693	-12.582
12	-0.398	-0.438	-0.483	9.329	17.656

propeller and rudder were used.

5.2.2 Turbulence modeling error

The level of grid refinement can affect the propeller and rudder forces, but the choice of turbulence models can also have a significant effect. In this study the Spalart Allmaras turbulence model and the $k-\omega$ SST turbulence model are used with wall functions. RANS CFD simulations with wall-resolved turbulence models, DES, or LES could be used to improve accuracy but these would require substantially increased computational cost. When separation occurs over a lifting foil and stall occurs there is a sudden loss in lift. Different turbulence models, the use of wall functions, and the grids used can influence the calculation of the separation point. *Eca et al.* (2018) illustrates how different turbulence models have different sensitivities to near wall spacing and even y^+ values of 1.0 can be insufficient.

Subsequently the prediction of the K_X, K_S, C_{Fx}, and C_{Fy} are discussed for LHS points five through twelve, correlating to the eight point LHS set. The G2 double-body grid is used. The percent error is calculated with respect to the Spalart Allmaras

prediction. Table 5.7 quantifies the difference between the prediction of K_S for the Spalart Allmaras turbulence model and the $k-\omega$ SST turbulence model for these LHS points for both the port and starboard side propellers. On average the starboard side is predicted more similarly between the two turbulence models compared to the port side. The average difference in terms of the absolute value for the prediction of K_X between the two turbulence models is only 1.61% on the starboard side, but 5.82% on the port side. This is sensible since the port propeller operates in the wake of the hull and this highlights how the interactions are calculated differently between the different turbulence models. LHS points seven and eight are closest to the experimental steady turn parameters of J and β. On the starboard side the K_X predicted by the $k-\omega$ SST turbulence model is 0.60% and 0.51% larger than the K_X predicted with the Spalart Allmaras turbulence model for LHS points seven and eight respectively. On the other hand the K_X predicted by the $k-\omega$ SST turbulence model is 10.20% larger and 6.57% smaller than the K_X predicted with the Spalart Allmaras turbulence model for LHS points seven and eight respectively. This illustrates the sensitivity of results to the turbulence model selection. For each of these LHS points the starboard side K_X is larger than the port side K_X when either turbulence model is used.

Table B.8 quantifies the difference between the prediction of K_S for the Spalart Allmaras turbulence model and the $k-\omega$ SST turbulence model for these LHS points for both the port and starboard side propellers. On average the starboard side is predicted more similarly between the two turbulence models compared to the port side. Table B.9 shows the differences between the two turbulence models prediction of C_{Fx} for each rudder.

Table 5.8 and Table B.10 show the differences between the two turbulence models prediction of C_{Fy} and C_{Mz} respectively for each rudder. The average absolute value of the difference between the C_{Fy} predictions between the two turbulence models is

Table 5.7: Turbulence modeling error between Spalart Allmaras and $k-\omega$ SST for K_X.

Side	LHS point	Spalart Allmaras (K_X)	$k-\omega$ SST (K_X)	ϵ_T (%)
Port	5	0.233	0.255	-9.659
Port	6	0.353	0.379	-7.392
Port	7	0.388	0.427	-10.200
Port	8	0.338	0.315	6.567
Port	9	0.194	0.184	5.400
Port	10	0.368	0.365	0.696
Port	11	0.434	0.433	0.252
Port	12	0.217	0.231	-6.360
Starboard	5	0.320	0.321	-0.220
Starboard	6	0.336	0.352	-4.704
Starboard	7	0.530	0.533	-0.604
Starboard	8	0.465	0.468	-0.512
Starboard	9	0.300	0.310	-3.450
Starboard	10	0.399	0.403	-0.914
Starboard	11	0.478	0.484	-1.405
Starboard	12	0.249	0.247	1.073

14.43% on the port side and 8.24% on the starboard side. Similarly, the average absolute value of the difference between the C_{Mz} predictions between the two turbulence models is 14.28% on the port side and 8.17% on the starboard side. The largest difference occurs at LHS point nine, for which the $k-\omega$ SST turbulence model prediction for the port side rudder has 31.21% more C_{Fy} than that predicted with the Spalart Allmaras turbulence model as well as a 30.28% larger C_{Mz}. This illustrates how lifting surfaces are sensitive to the turbulence models used and that modeling a lifting surface in the wake of an upstream body like a hull is challenging. A transitional turbulence model could also be examined and may be more accurate for determining the propeller and rudder forces due to the low Reynolds number on the propellers and rudders; however, using a transitional turbulence model would require finer wall-resolved grids.

Table 5.8: Turbulence modeling error between Spalart Allmaras and $k-\omega$ SST for C_{Fy}.

Side	LHS point	Spalart Allmaras (C_{Fy})	$k-\omega$ SST (C_{Fy})	ϵ_T (%)
Port	5	-0.325	-0.356	-9.581
Port	6	-0.443	-0.384	13.280
Port	7	-0.392	-0.365	7.011
Port	8	-0.359	-0.280	21.949
Port	9	-0.282	-0.370	-31.211
Port	10	-0.418	-0.380	9.174
Port	11	-0.441	-0.391	11.189
Port	12	-0.403	-0.452	-12.041
Starboard	5	-0.312	-0.337	-7.793
Starboard	6	-0.453	-0.465	-2.715
Starboard	7	-0.204	-0.214	-4.618
Starboard	8	-0.081	-0.073	9.824
Starboard	9	-0.219	-0.199	9.353
Starboard	10	-0.445	-0.472	-5.907
Starboard	11	-0.312	-0.351	-12.680
Starboard	12	-0.438	-0.495	-13.044

5.3 Data-Driven Model Development for ONR Tumblehome Propellers and Rudders

Different techniques are used to generate data-driven models models for each propeller and rudder. A set of different linear regression based data-driven models are trained and tested for different training sets. Linear regression models are generated and compared for the different grid densities for the training CFD simulations as well as for different turbulence models. Additionally, a Gaussian process regression model is developed for for the G2 double-body grid that uses the Spalart Allmaras turbulence model and model is compared to the linear regression based models.

5.3.1 Linear Regression Based Data-Driven Model Development

The regression based model is constructed using the methodology discussed in Chapter II. The form of the K_X and K_S are shown by Eqn. 2.26. The diameter of the propellers is 0.1066 m and the propeller revolution rate is 8.97 revolutions per second. The features used are those shown by Eqn. 2.25, which include J, J^2, β, β^2, and the cross coupling term $J\beta$. The linear regression model is generated using the MATLAB function `fitlm`. The regression model is trained with β in radians, however, for visualization purposes it is more natural to show β in degrees. The form of the rudder forces are shown by Eqn. 2.27. C_{Fx}, C_{Fy}, and C_{Mz} are calculated using Eqn. 2.15, Eqn. 2.16, and Eqn. 2.17 respectively for which the S_R is 0.012 m^2 and the d_r is 1.384 m. The resultant rudder velocity u_R for each propeller and the resulting β_R used in the rudder regression equation are calculated using Eqns. 2.19-2.12, where the K_X prediction using the model is used for the calculation of C_T. Fig. 5.8 shows the response surface of the K_X and K_S on both the port and starboard side propellers for the regression based data-driven model trained with the eight point LHS set with the G3 double-body grid. Fig. 5.9 shows the response surface of the C_{Fx} and C_{Fy} on both the port and starboard side propellers for the regression based data-driven model trained with the eight point LHS set with the G3 double-body grid.

5.3.2 Data-Driven Modeling Error Using Linear Regression

The data-driven model contains multiple sources of error including the discretization error and turbulence modeling error used to train and validate the model as well as the error in the model itself. The error inherent to the data-driven modeling technique used ϵ_r can be evaluated by how well the model predicts unseen validation data, for which LHS points one through four are used.

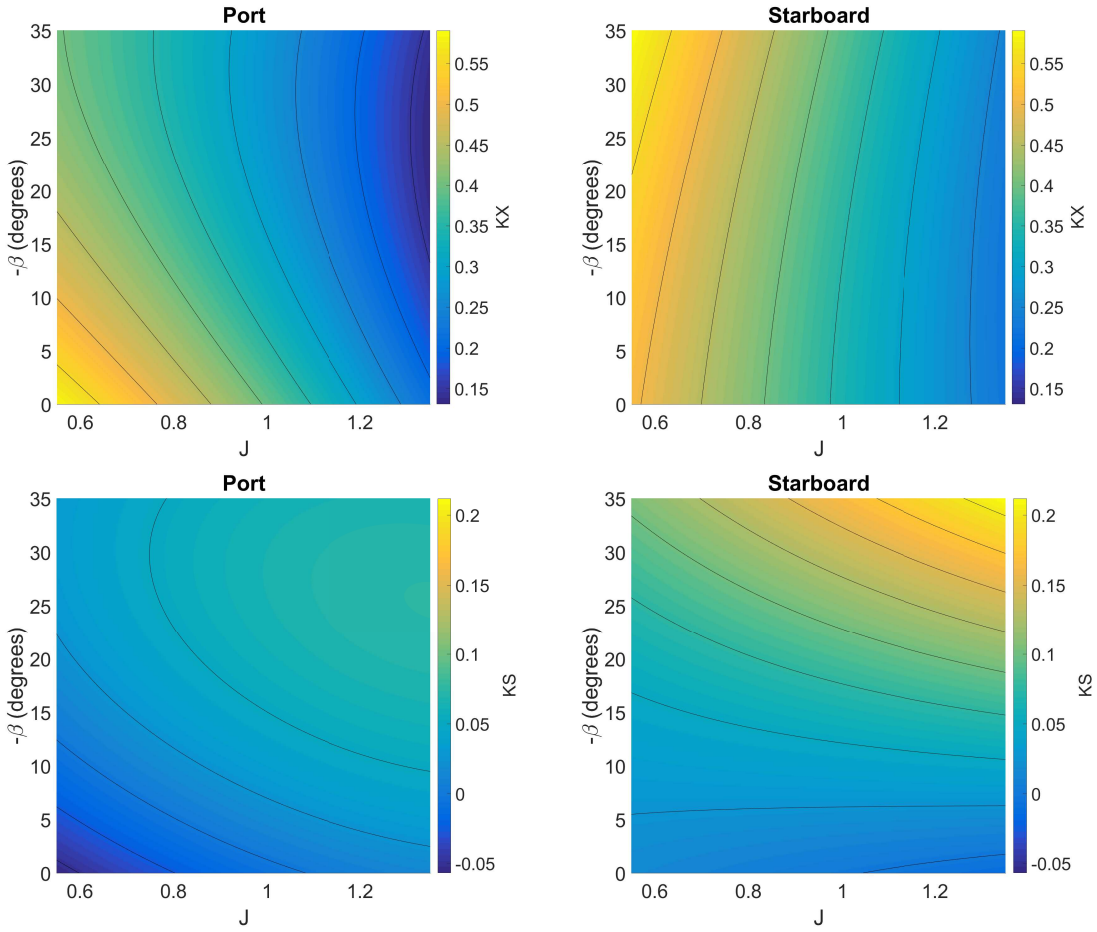

Figure 5.8: Response surface contours of force coefficients in J-β space for the model developed with the 8 point LHS set using the G3 training grid. K_X is shown at the top. K_S is shown on the bottom. The port side forces are shown on the left and the starboard forces are shown on the right.

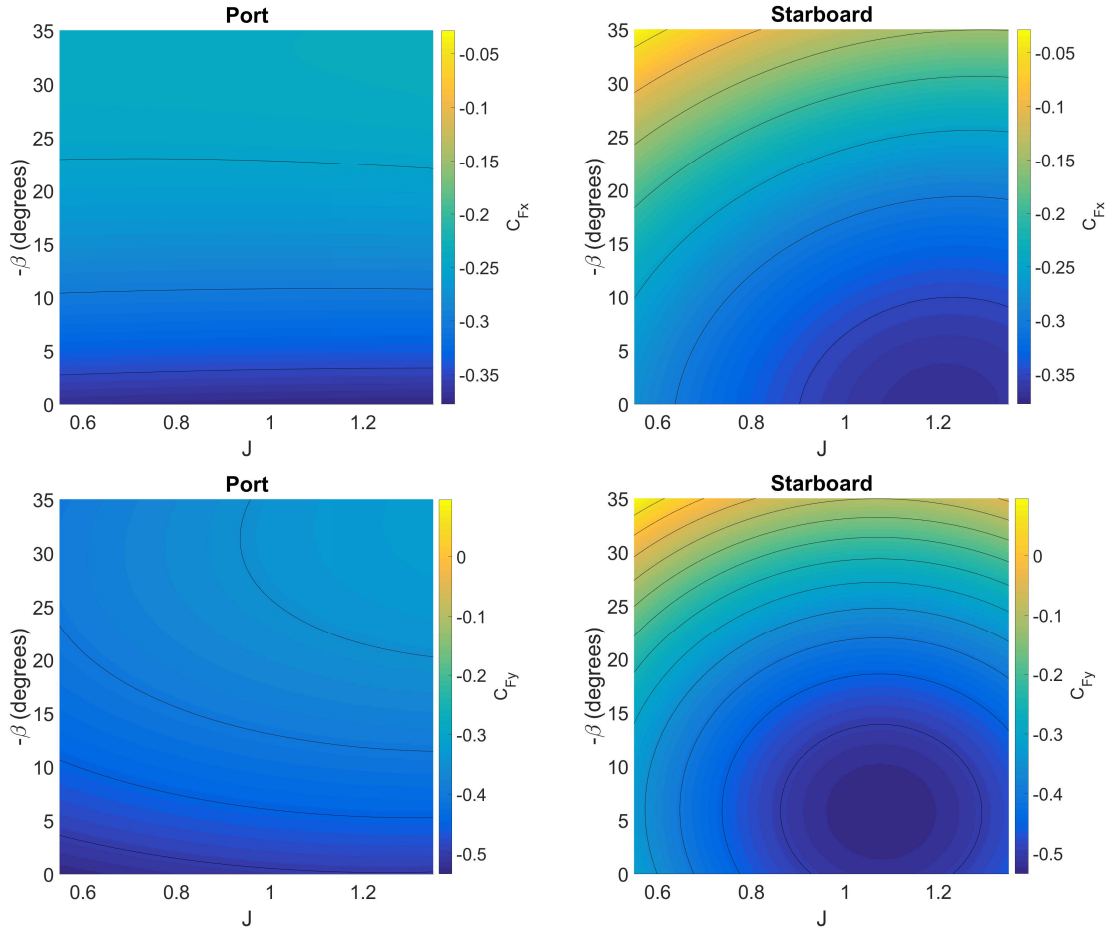

Figure 5.9: Response surface contours of force coefficients in J-β space for the model developed with the 8 point LHS set using the G3 training grid. C_{Fx} is shown at the top. C_{Fy} is shown on the bottom. The port side forces are shown on the left and the starboard forces are shown on the right.

Table 5.9: Regression modeling error ϵ_r for the G2 eight point LHS set with respect to LHS points 1-4.

Side	Parameter	point 1 (%)	point 2 (%)	point 3 (%)	point 4 (%)
Starboard	K_X	0.454	-0.579	-2.134	-2.572
Starboard	K_S	0.477	-2.602	0.402	-33.295
Starboard	C_{Fx}	-5.150	2.131	5.332	-1.405
Starboard	C_{Fy}	-9.916	-2.459	12.780	-5.476
Starboard	C_{Mz}	-9.521	-2.416	12.928	-5.534
Port	K_X	2.021	-3.828	6.348	7.054
Port	K_S	-0.913	5.401	-17.587	11.566
Port	C_{Fx}	3.847	2.000	2.774	0.875
Port	C_{Fy}	4.528	5.703	-2.377	2.994
Port	C_{Mz}	4.582	5.512	-2.168	2.891

Modeling Error with the Eight Point LHS Set on the G2 Grid Table 5.9 shows the error of the regression model prediction with respect to the CFD validation point prediction ϵ_r. The results are presented terms of percent error of the regression model prediction of each parameter for each propeller/rudder pair with respect to the CFD prediction for LHS points one through four. The regression model used here is trained with the eight point LHS set with the G2 double-body grid using the Spalart Allmaras turbulence model.

The average absolute value of the error ($\overline{|\epsilon_r|}$) for the K_X is 1.44% and 4.8% for starboard and port side propellers respectively. The $\overline{|\epsilon_r|}$ for the K_S is 9.19% and 8.67% for starboard and port side propellers respectively. On the starboard side propeller, the largest deviation between the CFD calculation and the data-driven model prediction is for LHS point four. For this point, the CFD prediction of K_X is 0.254 versus a data-driven model prediction of 0.260, which is the smallest magnitude of K_X of the four points examined here. Similarly, the K_S is very small with a CFD prediction of only 0.009, thus determining the error in terms of percent error skews the perception of the performance. Furthermore, this point correlates to a J of 1.25 and a β of 4.38°, thus it only has an influence on the maneuver when the rudder is

first actuated. On the port side propeller, the model error compared to the CFD prediction is largest for LHS point four with regards to the K_X prediction and is largest for LHS point three with regards to the K_S prediction with a magnitude of error of 7.05% and 17.59% respectively. LHS point three is also in the transient phase of the maneuver, whereas the predictions for LHS points one and two on both the starboard and port side are better predicted with the model.

The average magnitudes of the errors between the CFD and the data-driven model predictions for the rudder coefficients C_{Fx}, C_{Fy}, and C_{Mz} are 3.50%, 7.66%, and 7.6% respectively on the starboard side and 2.37%, 3.90%, and 3.79% on the port side. The largest percent error occurs for LHS point three on the starboard side for which the C_{Fx}, C_{Fy}, and C_{Mz} are 5.33%, 12.78%, and 12.93% respectively. Closer to the steady state operating point in the $J - \beta$ space, LHS point one has relatively large error with up to 9.92% error for the calculation of C_{Fy}, but for this point the starboard rudder has relatively low loading since it sees a clean inflow and a rather low level of incidence. Thus, the percent error is inflated since at this point the port side rudder has nearly three times more force and the model predicts this within 5% of the CFD calculation.

It is also important to consider the accuracy of the data-driven model in the context of the uncertainty in discretization error and turbulence modeling error. The K_X predicted by the model is within 1.5% and 5% of the CFD prediction on the starboard and port sides respectively, whereas the average discrepancy between the turbulence models examined with LHS points five through eight is in excess of 1.6% on the starboard side and 5.8% on the port side. Additionally, the average discretization error between the G2 and G3 grids for LHS points one through four using the Spalart Allmaras turbulence model is over 1.0% on the starboard side and greater than 1.6% on the port side. The uncertainty of the data-driven model prediction of the CFD calculation for the validation points is smaller than the sum of the discretization

uncertainty and turbulence modeling uncertainty.

Similarly, the K_S prediction of the data-driven model is within 9.2% on the starboard side and 8.9% on the port side. The average turbulence modeling discrepancy for LHS points five through twelve was found to be over 16% on the starboard side and over 18.7% on the port side. Additionally, the magnitude of the percent error between the G3 and G2 grids for LHS points one through four is over 6.9% on the starboard side and over 6.2% on the port side.

The rudder coefficients are also predicted accurately with the consideration of the uncertainty associated with the data that the model is trained with. The starboard side magnitude of the discrepancy between the G3 and G2 grids for the calculation of C_{Fx} and C_{Fy} using the Spalart Allmaras turbulence model for LHS points one through twelve is 7.6% and 14.9% respectively. Similarly, the average magnitude of the difference between the calculation of LHS points five through eight between the two turbulence models is nearly 5% for C_{Fx} and 8.2% for C_{Fy}. In comparison the data-driven model prediction has 3.5% and 7.6% error in terms of the average magnitude of the error of C_{Fx} and C_{Fy} respectively for the validation points, which is less than the uncertainty of both the discretization error and the turbulence modeling error.

The results for the port side rudder are similar. The port side magnitude of the discrepancy between the G3 and G2 grids for the calculation of C_{Fx} and C_{Fy} using the Spalart Allmaras turbulence model for LHS points one through twelve is 4.2% and 6.0% respectively. Similarly, the average magnitude of the difference between the calculation of LHS points five through eight between the two turbulence models is 10.6% for C_{Fx} and 14.6% for C_{Fy}. In comparison, the data-driven model prediction has only 2.4% and 3.9% error in terms of the average magnitude of the error of C_{Fx} and C_{Fy} respectively for the validation points, which is less than the uncertainty of both the discretization error and the turbulence modeling error.

Table 5.10: Regression modeling error ϵ_r for the G2 sixteen point LHS set with respect to LHS points 1-4.

Side	Parameter	point 1 (%)	point 2 (%)	point 3 (%)	point 4 (%)
Starboard	K_X	1.271	-1.380	-1.004	0.016
Starboard	K_S	0.833	-2.143	2.102	-19.337
Starboard	C_{Fx}	-11.414	7.346	3.973	-7.201
Starboard	C_{Fy}	-9.248	-1.994	12.493	-18.824
Starboard	C_{Mz}	-8.966	-1.773	12.596	-18.856
Port	K_X	5.989	-8.961	8.942	-4.561
Port	K_S	-5.973	-2.957	-23.151	-3.663
Port	C_{Fx}	4.973	-1.823	2.839	0.369
Port	C_{Fy}	6.129	3.683	-2.943	4.696
Port	C_{Mz}	6.114	3.431	-2.684	4.492

Modeling Error with the Sixteen Point LHS set on the G2 Grid Table 5.10 shows the ϵ_r in terms of percent error of the regression model prediction of each parameter for each propeller/rudder pair with respect to the CFD prediction for LHS points one through four. The regression model used here is that trained with the sixteen point LHS set with the G2 double-body grid using the Spalart Allmaras turbulence model.

The K_X on the starboard side is more accurate using the sixteen sample set instead of the eight point sample set with 0.9% error compared to 1.4% error respectively when the model prediction is compared to the four point LHS set of the CFD calculation using the same method that the model was trained with. Additionally, the starboard side K_S is more accurate when more samples are used with the average error reducing to 6.1% with the sixteen point LHS set compared to 9.2% with the eight point LHS set. On the other hand, the rudder force coefficients on the starboard side are less accurately predicted with the sixteen point LHS set predicting 7.5% error, 10.6% error, and 10.5% error for the average magnitude of the error for C_{Fx}, C_{Fy}, and C_{Mz} respectively; this is increased from the model which uses an eight point LHS set which predicted 3.5% error, 7.6% error, and 7.6% error for the average magnitude of the

error for C_{Fx}, C_{Fy}, and C_{Mz} respectively.

On the port side the average magnitude of the percent error of ϵ_r increases for each of the coefficients for the sixteen point LHS set. The average magnitude of the error for the port side K_X increases form 4.8% to 7.1%, the average magnitude of the error for the port side K_S increases form 8.87% to 8.94%, the average magnitude of the error for the port side C_{Fx} increases form 2.4% to 2.5%, the average magnitude of the error for the port side C_{Fy} increases form 3.9% to 4.4%, and the average magnitude of the error for the port side C_{Mz} increases form 3.8% to 4.2%. Overall, these changes are quite small, but it is unexpected that adding more samples leads to more error. One aspect that could affect this is that some of the samples in the sixteen point LHS set are far away from the operational space, notably for high J and high β.

Fig. 5.10 shows the different sources of error for the calculation K_X, K_S, C_{Fx}, and C_{Fy} for the G2 grid. The sources of error are the discretization error, the uncertainty due to turbulence modeling error, and the error of using the eight point set versus the sixteen point set for training the regression model. The discretization error is calculated by taking the average of the error between the G2 and G3 grid for each coefficient ($\overline{|\epsilon_{23}|}$). The turbulence modeling uncertainty is the average of the absolute value of the difference between the predictions of each sample point between the two turbulence models examined ($\overline{|\epsilon_T|}$). The `8 Point LHS Model` error is the $\overline{|\epsilon_r|}$ when the model is trained with the eight point LHS set and validated with regards to the CFD prediction of the four point validation data set; likewise the `16 Point LHS Model` error is the $\overline{|\epsilon_r|}$ when the model is trained with the sixteen point LHS set and validated with regards to the CFD prediction of the validation data set. All sources of error for the prediction of K_X are less than 6%. The largest average error is on the port side. The turbulence modeling uncertainty dominates. The linear regression model predicts the starboard side K_X well, but the $\overline{|\epsilon_r|}$ on the port side is nearly as large as the underlying turbulence modeling uncertainty. For the prediction of K_S, the

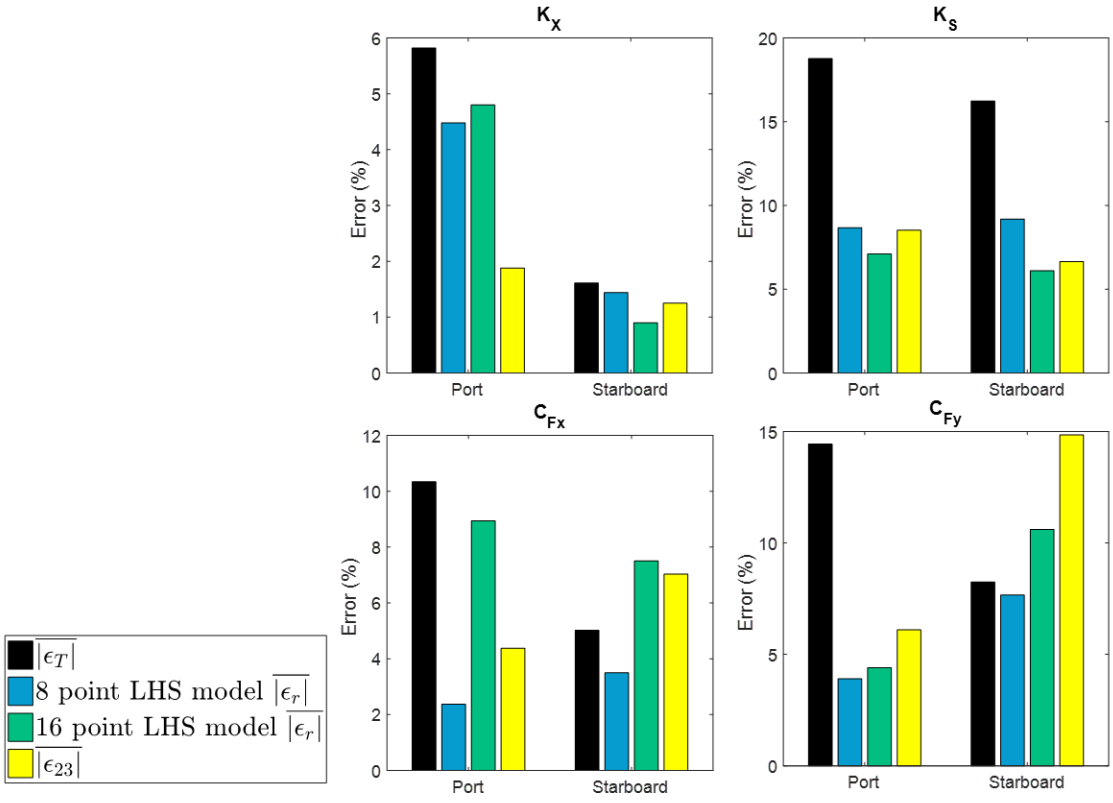

Figure 5.10: Average error for K_X (upper left), K_S (upper right), C_{Fx} (lower left) and C_{Fy} (lower right) for G2 based linear regression models. This summarizes the underlying discretization error and turbulence modeling error in the context of how well the linear regression models predict the CFD validation data.

turbulence modeling uncertainty is again dominant, and the error induced by using the linear regression models is on the same order as the underlying discretization error. For the rudder force coefficients, the linear regression model trained with the eight point set outperforms the model trained with the sixteen point set. The discretization error on the starboard side is large, especially for the prediction of C_{Fy}. The turbulence modeling error on the port side, which is the rudder that operates in the wake of the vessel is also large. The error induced by the linear regression models is less than underlying uncertainty due to discretization error and turbulence modeling uncertainty.

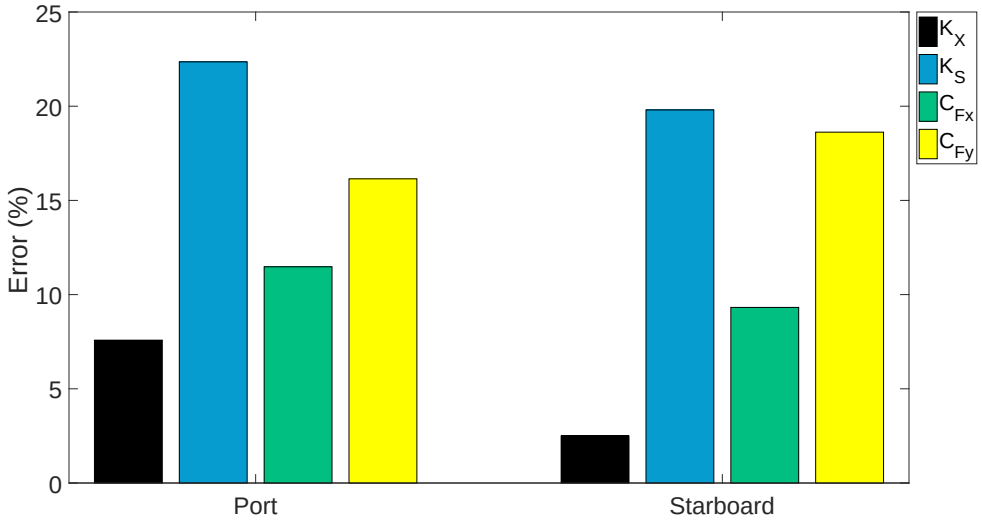

Figure 5.11: $\overline{|\epsilon|}$ for the different force coefficients based on the G2 based linear regression model trained with the eight point sample.

To further characterize the error, each component of the error including the discretization error, the turbulence modeling error, and the regression modeling error can be combined into an error metric $\overline{|\epsilon|}$ as defined earlier by Eqn. 3.7. The $\overline{|\epsilon|}$ for the K_X, K_S, C_{Fx}, C_{Fy} is shown by Fig. 5.11 for both the port and starboard sides.

5.3.3 Effect of Training Data on Linear Regression Data-Driven Model Predictions

One way to examine the effect of training data in more detail is to compare the differences in the response surfaces themselves. This is useful to do for different LHS sets for training the models, different turbulence models, and different grids. Additionally, the various models generated with different sets of data are implemented for the vessel performing maneuvers and the differences between the models can be quantified in terms of the differences between the different maneuvering characteristics. Here, the differences between the response surfaces generated with the eight point LHS set using the G2 grid are compared if the data used utilizes the Spalart

Allmaras turbulence model in the CFD training versus the $k-\omega$ SST turbulence model. Fig. 5.12 compares the port and starboard side propellers for the prediction of K_X. In the figure, the response surface generated using the CFD simulations that use the Spalart Allmaras turbulence model are shown on the top, the response surfaces generated using the CFD simulations that use the $k-\omega$ SST turbulence model for training are shown in the middle, and the percent difference between the two, with respect to the Spalart Allmaras response surface is shown on the bottom. The left hand side shows the plots for the port side propeller, while the plots on the right show the results for the starboard side propeller. While the forms of the two models are quite similar overall, the curvature of the contour lines is different between the two models. The differences on the starboard side propeller are quite low, but the port side propeller has more significant differences, especially for high values of J and β, however this region of the plot is outside of the operating space of the vessel.

Fig. 5.13 compares the port and starboard side rudders for the prediction of C_{Fy}. The response surface generated using the CFD simulations that use the Spalart Allmaras turbulence model are shown on the top, the response surfaces generated using the CFD simulations that use the $k-\omega$ SST turbulence model for training are shown in the middle, and the difference between the two, with respect to the Spalart Allmaras response surface is shown on the bottom. The left hand side shows the plots for the port side rudder, while the plots on the right show the results for the starboard side rudder. The form of the response surface on the starboard side is quite similar, with the largest difference being the minimum value of C_{Fy}, for which the magnitude is largest when $|\beta|$ is low so the rudder operates at a high angle of incidence. On the port side, the form of the surrogate model is different. For the model generated with the Spalart Allmaras turbulence model based CFD training points, the largest magnitude force is when $|\beta|$ is low and J is low and the magnitude of C_{Fy} decreases as each of the parameters J and $|\beta|$ increase. On the other hand, the model gener-

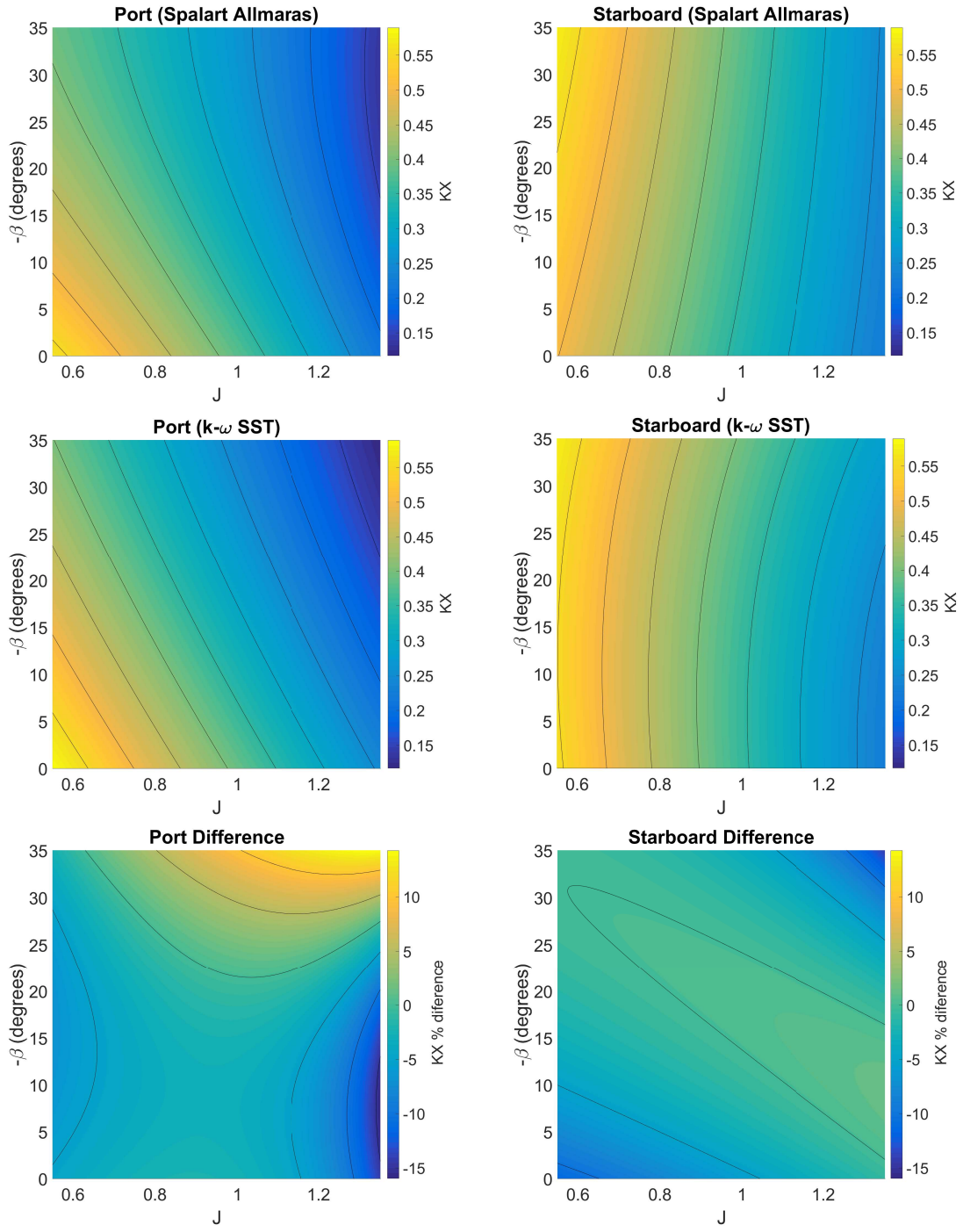

Figure 5.12: Contours of K_X in J-β space. The K_X predicted with the eight point LHS using the Spalart Allmaras turbulence model is shown at the top. The K_X predicted with the eight point LHS using the $k-\omega$ SST turbulence model is shown in the middle. The percent difference of the K_X predicted by the two models with respect to the eight point LHS using the Spalart Allmaras turbulence model is shown on the bottom. The port side forces are shown on the left and the starboard forces are shown on the right.

ated with the training points that use the $k-\omega$ SST turbulence model has the lower magnitude of C_{Fy} at high $|\beta|$, but at a lower J.

5.3.4 Gaussian Process Regression Based Data-Driven Model Development

The MATLAB function `fitrgp` is used to train, validate, and examine the response surfaces generated with different amounts of training data based on the CFD results from the G2 double-body grid with the Spalart Allmaras turbulence model. The four point LHS set is used to validate the model and to compare the performance versus the linear regression predictions. The eight point LHS set and the sixteen point LHS set are used to train the GPR models for each force coefficient. Three different sets of training are examined: the eight-point LHS set, the sixteen-point LHS set, and the combination of both the eight point and sixteen point sets to generate a twenty-four point training set.

As σ_n approaches zero the prediction of the GPR at the training points improves. Thus, when σ_n is zero, the training point values are exactly predicted. This is a good assumption if the data used to train the model is high fidelity and it is safe to assume that any noise in the GPR prediction is not caused by spurious data points that are used to train the model. If on the other hand, there are spurious training points, then the model over-fits the prediction based on training data that may not be correct.

Two different approaches are used to generate GPR models. The first approach assumes that the noise in the training data is near zero ($\sigma = 0.0001$). The second approach calculates the noise in the training and validation data by determining the standard deviation of the uncertainty.

σ_n Near-zero First the effect of using a very small σ_n of 0.0001 is characterized. With this small value of noise, the GPR fits very accurately to the data that it is

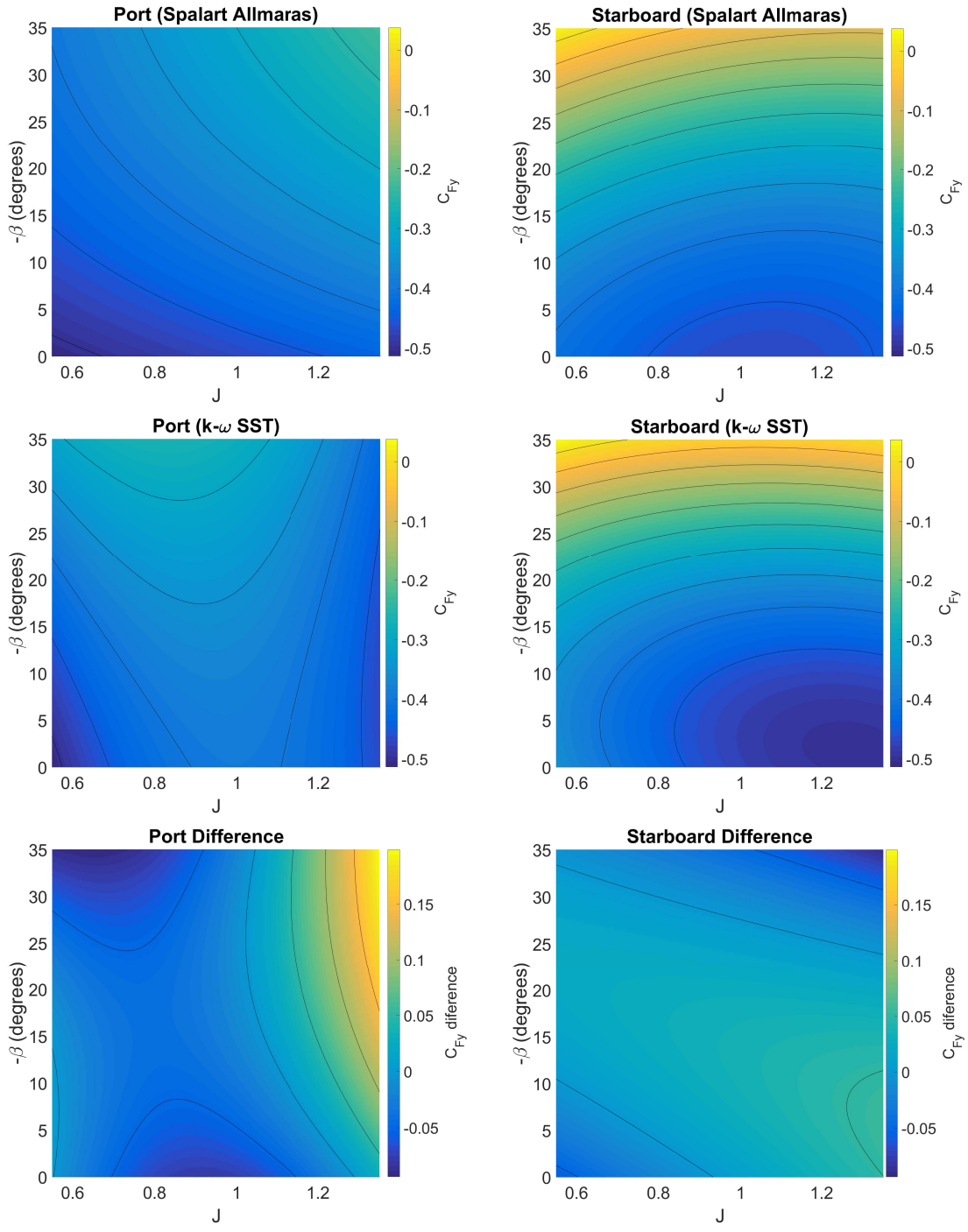

Figure 5.13: Contours of C_{Fy} in J-β space. The C_{Fy} predicted with the eight point LHS using the Spalart Allmaras turbulence model is shown at the top. The C_{Fy} predicted with the eight point LHS using the $k-\omega$ SST turbulence model is shown in the middle. The difference of the C_{Fy} predicted by the eight point LHS using the Spalart Allmaras model minus the C_{Fy} predicted with model trained with the $k-\omega$ SST turbulence model is shown on the bottom. The port side forces are shown on the left and the starboard forces are shown on the right.

trained with. Fig. 5.14 shows the K_X for the propellers trained with the eight point sample set (in the top row), the 16 point sample set (in the middle row), and the combination of the 16 point and 8 point sample set (in the bottom row). The left hand column of images correlates to the response surface for the port side propeller, while the right hand column corresponds to the response surface for the starboard side propeller. The black dots in this figure and in subsequent figures illustrates the training points used. When only eight points are used to generate the response surface, the contours are rather smooth. On the other hand, when the number of samples is increased, the response surface becomes less smooth because the response surface conforms to each of the training points. Using the GPR prediction when σ_n of 0.0001 is used, adding more training points leads to an improved prediction of the four validation points based on the results of LHS points one through four. The average percent error of the GPR prediction with relation to the four validation points calculated with CFD for the K_X of the port side propeller is 6.47%, 3.86%, and 1.74% respectively for the eight point LHS set, the sixteen point LHS set, and the combination of both the eight and sixteen point LHS sets for training the model; for the starboard side prediction of K_X, the GPR predicts with 1.18%, 3.71%, and 0.72% respectively. It is notable that the sixteen point set of training data leads to a worse prediction than the eight point set of data for the starboard side propeller; however, when all twenty four points of data are used the accuracy of the GPR model improves. This illustrates that the GPR is able to predict the results for the model that it is trained with better as more training data is used. It is also noteworthy that the linear regression model trained with the eight point LHS set more accurately predicted the port side propeller K_X, but predicted the starboard side K_X slightly less accurately. Conversely, the linear regression model trained with sixteen points outperforms the GPR with sixteen training points for the starboard side K_X, but the port side K_X is better predicted with the GPR when the same training set is used. It

is important to note that the response surface has more variation than what would be expected for the propeller K_X, which intuition suggests should produce a smoother response surface, and that this variation could be due to the noise or uncertainty in the CFD calculations.

Fig. B.1 shows the K_S for the propellers trained with the eight point sample set, the 16 point sample set, and the combination of the 16 point and 8 point sample set for a σ_n of 0.0001. The average percent error of the GPR prediction with relation to the four validation points calculated with CFD for the K_S of the port side propeller is 10.36%, 4.28%, and 6.90% respectively for the eight point LHS set, the sixteen point LHS set, and the combination of both the eight and sixteen point LHS sets for training the model; for the starboard side prediction of K_S, the GPR predicts with 10.80%, 8.46%, and 9.83% respectively. The linear regression model outperforms the GPR in terms of accuracy for the port side for both the eight point and sixteen point LHS sets, as well as on the starboard side for the eight point LHS set, but not the sixteen point LHS set.

Fig. B.2 shows the C_{Fx} for the rudders trained with the eight point sample set, the 16 point sample set, and the combination of the 16 point and 8 point sample set for a σ_n of 0.0001. The average percent error of the GPR prediction with relation to the four validation points calculated with CFD for the C_{Fx} of the port side rudder is 3.86%, 1.60%, and 2.09% respectively for the eight point LHS set, the sixteen point LHS set, and the combination of both the eight and sixteen point LHS sets for training the model; for the starboard side prediction of the rudder C_{Fx}, the GPR predicts with 4.68%, 10.42%, and 2.61% accuracy respectively.

Fig. B.3 shows the C_{Fy} for the rudders for a σ_n of 0.0001. The average percent error of the GPR prediction with relation to the four validation points calculated with CFD for the C_{Fy} of the port side rudder is 2.71%, 3.02%, and 2.89% respectively for the eight point LHS set, the sixteen point LHS set, and the combination of both

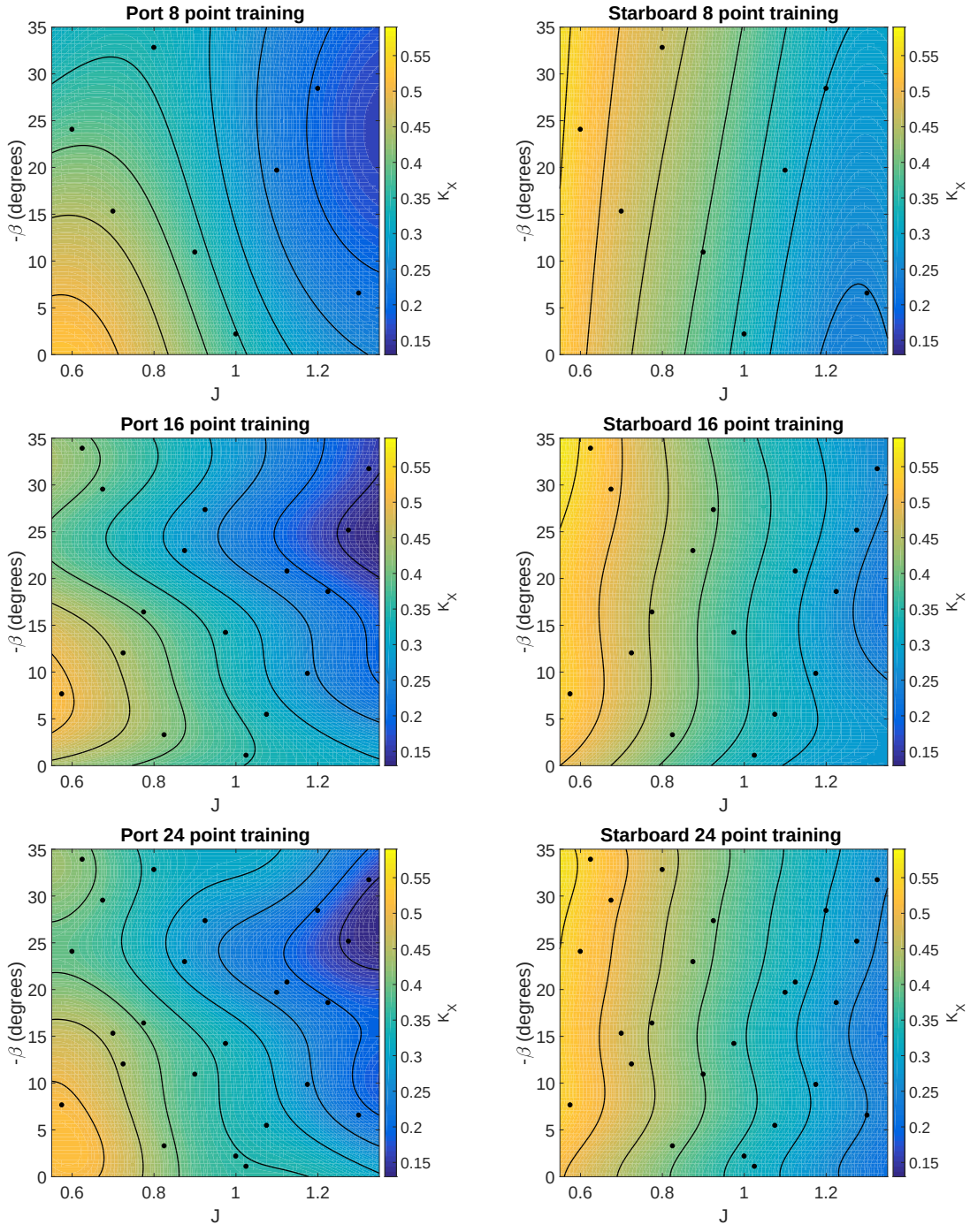

Figure 5.14: Contours of K_X in J-β space predicted with Gaussian process regression with σ_n of 0.0001. The model using the 8-point LHS set is on the top. The model using the 16-point LHS set is in the middle row. The model using the combination of the 8-point and 16-point LHS set is on the bottom. The port side forces are shown on the left and the starboard forces are shown on the right.

the eight and sixteen point LHS sets for training the model; for the starboard side prediction of the rudder C_{Fy}, the GPR predicts with 10.30%, 8.45%, and 4.36% accuracy respectively. The linear regression model outperforms the GPR prediction of the validation data for the the prediction of C_{Fx} on both the port and starboard side when the eight point set of training data is used to train the models. Additionally, the linear regression model outperforms the GPR prediction of C_{Fx} on the starboard side for the sixteen point set of training data, but not on the port side. For the prediction of C_{Fy} with the eight point set of training data, the linear regression model better predicts the starboard force, but the GPR predicts the port side force better. The C_{Fy} for the sixteen point set of data is better predicted on both port and starboard side by the GPR model compared to the linear regression model prediction. From comparing the prediction of the validation data using both the linear regression and the GPR, one is not substantially more accurate than the other.

In Gaussian process regression, the standard deviation of the noise σ_n can be accounted for in the generation of the model. If σ_n is zero, then the Gaussian process regression predicts the sample points exactly and the training error would be zero as is the case with traditional Kriging models *Liem et al.* (2015). If on the other hand, σ_n is calculated based on the uncertainty of the data, the training points are no longer treated as truth such that the error at the training points is non-zero, but the response surface becomes smoother and does not over-fit the noisy training data as is illustrated next.

σ_n Calculated The σ_n is calculated as a function of the discretization error and the turbulence modeling error. The standard deviation of the discretization error is calculated from LHS points one through twelve with the Spalart Allmaras turbulence model, and the discretization error is approximated by the difference between the G2 grid and the G3 grid prediction. The effect of the turbulence modeling error is taken

by examining the eight sample points for LHS points five through twelve for which the G2 grid was run with both turbulence models. The standard deviation of the turbulence modeling error is calculated using the difference in the calculation across these eight samples. The total standard deviation is calculated as the sum of both the discretization error standard deviation and the turbulence modeling error standard deviation. This σ_n is calculated for each force coefficient.

Fig. 5.15 shows the K_X for the propellers for a calculated σ_n of 0.037 for the port side propeller and for 0.010 for the starboard side propeller. The image layout for the figure is the same as the previous set of figures with the number of training sample points increasing with each row, and with the port and starboard side coefficients on the left and right respectively. A smooth response surface is seen for all sets of training. The average percent error of the GPR prediction with relation to the four validation points calculated with CFD for the K_X of the port side propeller is 5.90%, 6.10%, and 5.66% respectively for the eight point LHS set, the sixteen point LHS set, and the combination of both the eight and sixteen point LHS sets for training the model; for the starboard side prediction of the propeller K_X, the GPR predicts with 0.96%, 1.11%, and 1.15% accuracy respectively. Calculating σ_n leads to an improvement for the sparsest sampled training set of data, but leads to worse predictions of the validation data for training data sets that use more samples. This again illustrates how when σ_n is assumed small it can lead to over-fitting of data, but with enough samples it can predict unseen data points well. However, if there is uncertainty in the data, using a calculated σ_n leads to a smoother prediction, that may not predict the validation data as accurately, but if the validation data is not the truth because it contains discretization and turbulence modeling uncertainty, it makes sense to not over-fit the model to noisy data.

Fig. B.4 shows the K_S for the propellers for a calculated σ_n of 0.022 for the port side propeller and for 0.010 for the starboard side propeller. The average percent error

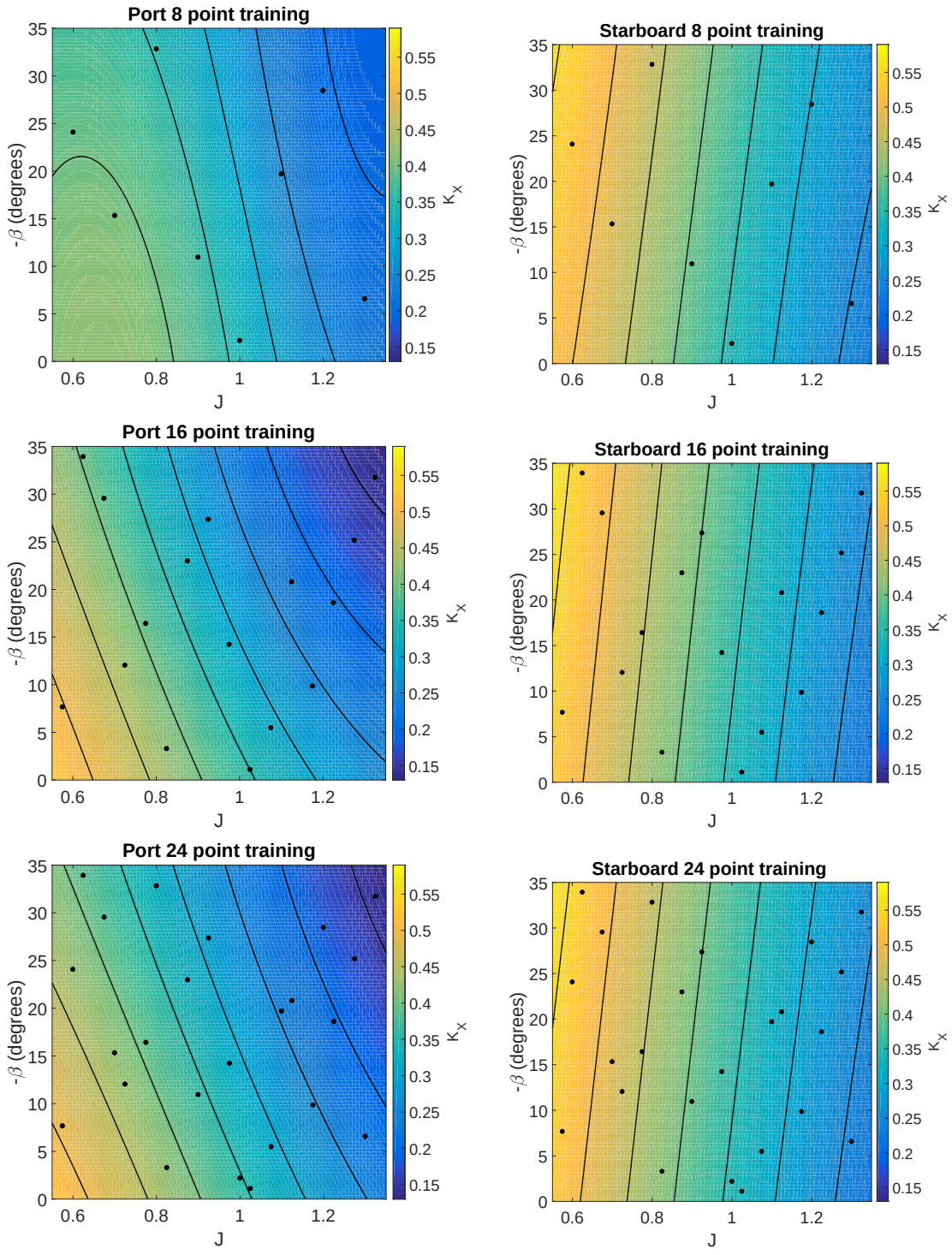

Figure 5.15: Contours of K_X in J-β space predicted with Gaussian process regression with σ_n calculated. The model using the 8-point LHS set is on the top. The model using the 16-point LHS set is in the middle row. The model using the combination of the 8-point and 16-point LHS set is on the bottom. The port side forces are shown on the left and the starboard forces are shown on the right.

of the GPR prediction with relation to the four validation points calculated with CFD for the K_S of the port side propeller is 6.36%, 6.94%, and 6.18% respectively for the eight point LHS set, the sixteen point LHS set, and the combination of both the eight and sixteen point LHS sets for training the model; for the starboard side prediction of the propeller K_S, the GPR predicts with 24.10%, 25.07%, and 17.43% accuracy respectively. Accounting for the noise in CFD calculations leads to an improvement in the port side K_S for the eight point sample, but otherwise leads to worse predictions relative to the CFD calculations.

Fig. B.5 shows the C_{Fx} for the propellers for a calculated σ_n of 0.057 for the port side rudder and for 0.023 for the starboard side rudder. The average percent error of the GPR prediction with relation to the four validation points calculated with CFD for the C_{Fx} of the port side rudder is 6.14%, 2.89%, and 3.21% respectively for the eight point LHS set, the sixteen point LHS set, and the combination of both the eight and sixteen point LHS sets for training the model; for the starboard side prediction of the C_{Fx}, the GPR predicts with 4.87%, 6.23%, and 5.12% accuracy respectively. Accounting for the noise in CFD calculations leads to an improvement in the starboard side C_{Fx} for the model trained with sixteen sample points, but otherwise leads to worse predictions relative to the GPR model that assumed no noise. In general, the GPR model with the calculated σ_n performs worse at predicting the validation CFD data points on average compared to assuming that there is no noise in the inputs. Fig. B.6 which shows the C_{Fy} for the propellers for a calculated σ_n of 0.106 for the port side rudder and for 0.045 for the starboard side rudder. The average percent error of the GPR prediction with relation to the four validation points calculated with CFD for the C_{Fy} of the port side rudder is 8.55%, 5.12%, and 4.86% respectively for the eight point LHS set, the sixteen point LHS set, and the combination of both the eight and sixteen point LHS sets for training the model; for the starboard side prediction of the C_{Fy}, the GPR predicts with 10.67%, 8.59%, and 7.56% accuracy respectively. When

noise is accounted for, the GPR performs worse at predicting the validation points than when the noise is assumed to be negligible. Additionally, it is noteworthy that on the port side, the variation in the prediction of C_{Fy} is washed out since the noise of the uncertainty in the CFD data dominates, especially for sparsely sampled data.

Overall, when the uncertainty of the CFD data is accounted for by calculating σ_n, the benefit of using GPR over linear regression is small and can lead to over-fitting of noisy data. With uncertain data, the GPR predicts smooth responses that are qualitatively similar to the linear regression response surface. If the fidelity of the inputs used to train and validate the model are high, then the GPR could be a very useful tool since the noise in the response surface could be assumed true, however, the uncertainty of the GPR with uncertain inputs is large. For example the uncertainty response surface can be plotted for K_X. Fig. 5.16 shows the breadth of the 95% confidence interval of K_X on the port side propeller (Δ_{Kx}) plotted in terms of the percent value of the local K_X for each respective J-β. Eqn. 5.1 shows how Δ_{Kx} is calculated in terms of the upper bound of K_X provided by the GPR uncertainty K_{XU}, and the lower bound of the K_X provided by the GPR uncertainty K_{XL}. The Δ_{Kx} non-dimensionalized by the K_X predicted by the model is shown for each set of training data for the GPR and the left hand column of images correlates to a σ_n of 0.0001, while the right hand column of images correlates to a calculated σ_n of 0.037. This image illustrates that when σ_n is zero, the uncertainty predicted by the GPR at the training points is very small. On the other hand, if σ_n is non-zero, then the GPR predicts an uncertainty at the training points, since the data itself is uncertain. As more training data is added the uncertainty for the GPR with low σ_n becomes small across the whole J-β space. On the other hand, even with lots of training data, the case with calculated non-zero σ_n is large. When σ_n is calculated there are no longer regions of low uncertainty around the training points. Also it is important to note, that in the upper right hand corner of the plot, the error is washed out due to the

scale used, but this is outside of the operating J-β space of the vessel.

$$\Delta_{Kx} = K_{XU} - K_{XL} \tag{5.1}$$

Depending upon the fidelity of data that is available for training a model different data-driven modeling techniques may be desirable. In this study, since the CFD uncertainty is not small, linear regression is a better fit for implementation in the CFD analysis of a maneuvering vessel. However, a practitioner of this method could use GPR if the fidelity of the training data is very high and a large number of samples are available such that the fidelity of both the data to train the model and the model itself is very high. On the other hand, linear regression has been shown to do a good job of predicting the validation data and is easier to implement in a custom CFD package.

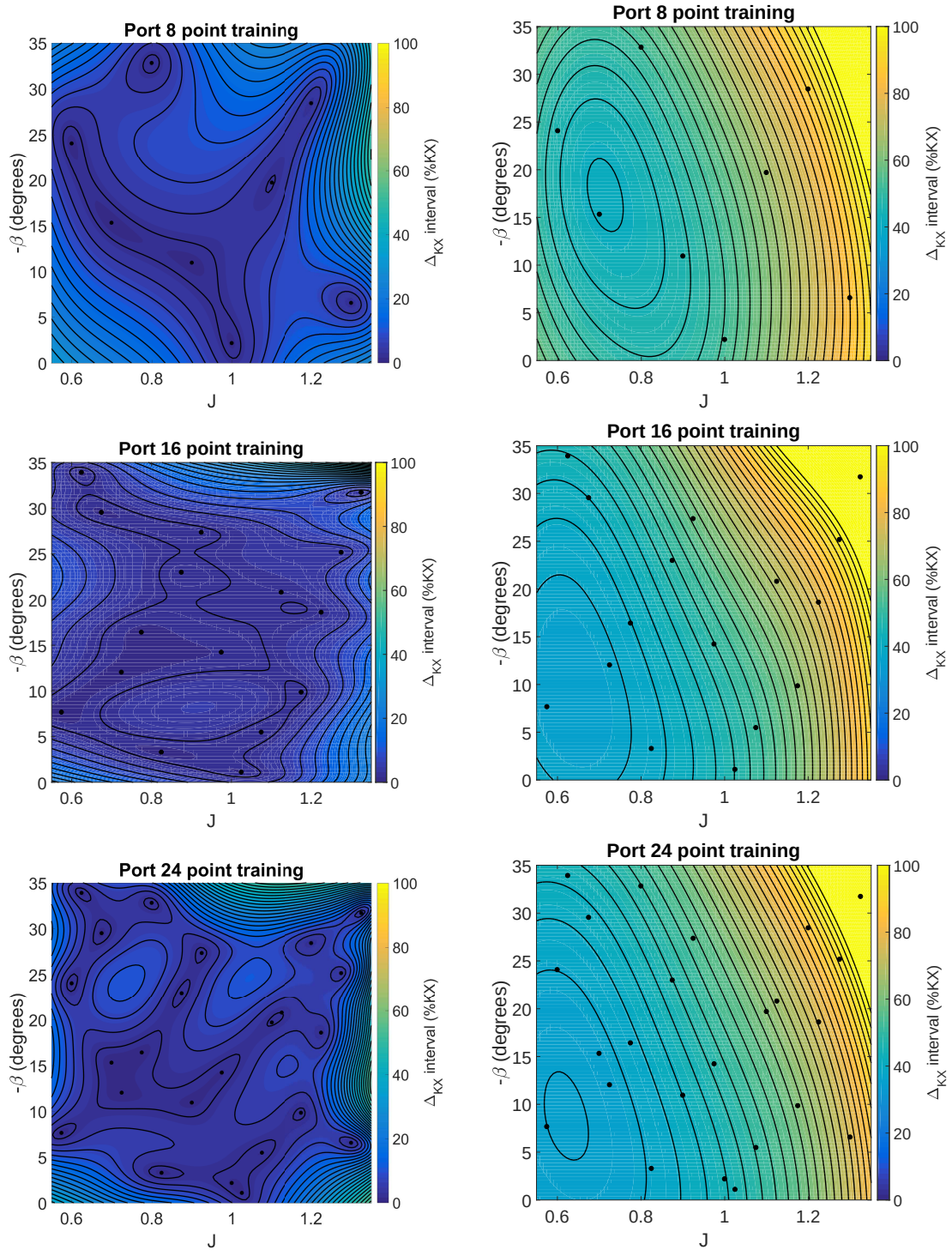

Figure 5.16: Contours of Δ_{Kx} in terms of a % of the local K_X is shown in J-β space predicted with Gaussian process regression. The model using the 8-point LHS set is on the top. The model using the 16-point LHS set is in the middle row. The model using the combination of the 8-point and 16-point LHS set is on the bottom. The response surface with low value of σ_n is shown on the left and the response surface with calculated σ_n is shown on the right.

CHAPTER VI

Maneuvering of the ONR Tumblehome

This chapter investigates the ONR Tumblehome performing turning circle maneuvers both in calm water and in waves. Various linear regression based data-driven propeller and rudder models, differing from each other by the training data used, are examined and applied to VOF CFD maneuvering simulations of the ONR Tumblehome. A discussion of the CFD and data-driven model setup, the maneuvering characteristics are examined at a Froude number of 0.2 for turning circles towards port with the rudder deflected to -35° in both calm water and waves. These maneuvers are compared to experiments (*SIMMAN* (2020). The importance of the propeller side force on the maneuvering characteristics of the vessel. The ability for the model to scale is also examined by examining the maneuvering characteristics of the vessel at Froude numbers of 0.1, 0.2, and 0.3 relative to the experimental results provided by *Sanada et al.* (2019).

6.1 VOF CFD Model Development

OpenFOAM version 2.4.x is used for the analysis. The maneuvering CFD simulations use the VOF approach to capture the interface between the water and air and the data-driven model is used to determine the force of both propellers and both rudders (including both the stationary and moving parts). The VOF grid is constructed

in a similar way as the double-body grid that is used to train the data-driven models (in Chapter V) with the main exception that the air domain is also modeled. Additionally, the geometry of the propellers and rudders are removed from the simulation and are instead modeled with the data-driven model. The body force of both the propellers and both of the rudders are applied to the flow via a body force region. The body force is important to include since it not only applies a velocity source at the stern of the vessel which leads to the thrust deduction effect, but since the force is asymmetric between the port and starboard sides, an asymmetric force acts on the hull which can change the maneuvering characteristics. The body force regions for the propellers are the the cylinders that encompasses the swept volume of each propeller. The body force region for the rudders are the bounding boxes that encompasses each rudder. The force on each propeller and rudder is calculated individually with the linear regression based data-driven model and the body force is applied uniformly across the respective zone in the lateral and axial directions only.

Several grid densities are examined based off of the same grid structure as the G1 and G2 double body grids with the air region added. The region around the vessel has isotropic cells and stretching is applied between this region and each of the boundaries at the edge of the domain. The the effect of refining the region of the waterline is examined by refining the Z direction by a factor of $\sqrt{2}$, and by a factor of 2. In total five grids are examined, these include: a `G1` VOF grid, a G1 VOF grid with the waterline region refined by a factor of $\sqrt{2}$ (denoted `G1` $\Delta z/\sqrt{2}$), a G1 VOF grid with the waterline region refined by a factor of two (denoted `G1` $\Delta z/2$), a `G2` VOF grid, and a G2 VOF grid with the waterline region refined by a factor of $\sqrt{2}$ (denoted `G2` $\Delta z/\sqrt{2}$). The grid size as well as the dimensions of each cell (Δ_g) in the region around the vessel for each of these grids is shown in Table 6.1. One level of refinement is applied to the hull and two levels of refinement are applied to the bulb, bilge keels, propeller shaft, and struts. By refining the waterline, the mesh density

Table 6.1: Grid size for ONR Tumblehome VOF simulations.

Grid	Grid size (million cells)	Δ_g (m)	Δ_{gx} (m)	Δ_{gy} (m)	Δ_{gz} (m)
G1	2.15	0.028	0.028	0.027	0.028
G1 $\Delta z/\sqrt{2}$	2.25	0.025	0.028	0.027	0.020
G1 $\Delta z/2$	2.37	0.023	0.028	0.027	0.014
G2	5.61	0.020	0.020	0.020	0.020
G2 $\Delta z/\sqrt{2}$	5.91	0.018	0.020	0.020	0.014

at the interface between the water and air can be better captured, by refining the waterline on the G1 grid by a factor of $\sqrt{2}$ the vertical grid density at the waterline is the same as the G2 grid, and when the waterline is refined by a factor of 2 the vertical grid density is the same as the vertical refinement of the G3 double body grid.

Fig. 6.1 shows the grid along the calm waterline and along the centerline of the vessel. This figure also shows the domain extents with the light gray surfaces. The boundary conditions on the hull, bulb, struts, bilge keels, and propeller shafts are walls with wall functions. The Spalart Allmaras turbulence model is used unless otherwise specified. The surfaces upstream, downstream, at the bottom of the domain, towards port, and towards starboard are all wave inlets. The top of the domain is a pressure inlet-outlet. A customized waves2Foam (*Jacobsen et al.* (2011)) package is used for the VOF simulations. A cylindrical wave relaxation zone is specified with an inner radius of 1 L_{WL} from the center of gravity of the vessel, and a maximum radius corresponding to the maximum extents of the domain. For calm water cases a potentialCurrent wave type is used with zero velocity. For cases with ambient waves, the waves are specified as regular fifth-order Stokes waves.

For each simulation a customized six degree of freedom rigid body motion solver is used. The vessel is accelerated from rest to the specified Froude number over eight seconds of model scale time. During this ramp the vessel is free to pitch and heave. Once the vessel accelerated to the desired speed, the speed is held constant for four seconds before it is released and all six degrees of freedom are solved for.

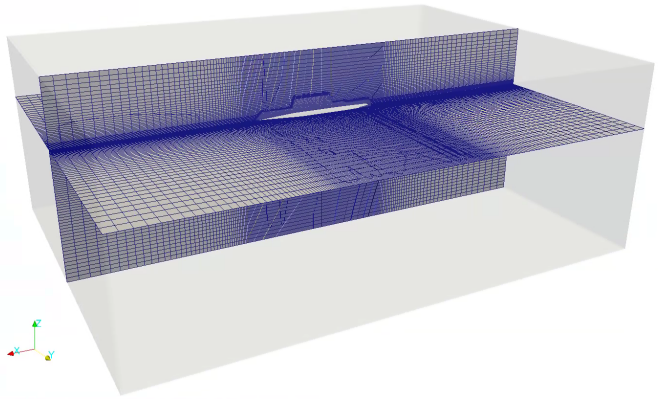

Figure 6.1: ONR Tumblehome G1 $\Delta z/2$ VOF grid.

The data-driven propeller and rudder model apply the body force to the flow and the forces to the equations of motion to perform the desired maneuvers. In this study, the force predicted by the data-driven model is applied assuming the rudder angle is deflected to the maximum deflection angle for the duration of the simulation. In the experiment the rudder deflection rate was 35° per second, thus it takes one second for the propeller to deflect from the deflected position to the maximum deflected position. This transient is ignored in this study, and instead when the CFD results are compared to the experiment the time and position are adjusted by 0.5 seconds, or the time that it would take for the propeller to reach its midway deflection point. Alternatively, the forces could be interpolated between the prediction of the data-driven model of the force with a deflected rudders versus a calculation of the propeller and rudder forces with the rudders undeflected ($\delta = 0°$). The simulations ramp up to the initial forward velocity of the vessel from rest over eight seconds and the model is held at constant forward velocity four an additional four seconds before the vessel is released. During the ramp of forward velocity, the vessel is free to heave and pitch, but the other degrees of freedom are constrained.

Different variations of the linear regression based data-driven propeller and rudder model are used to apply the forces of the propeller and rudder to the equations of

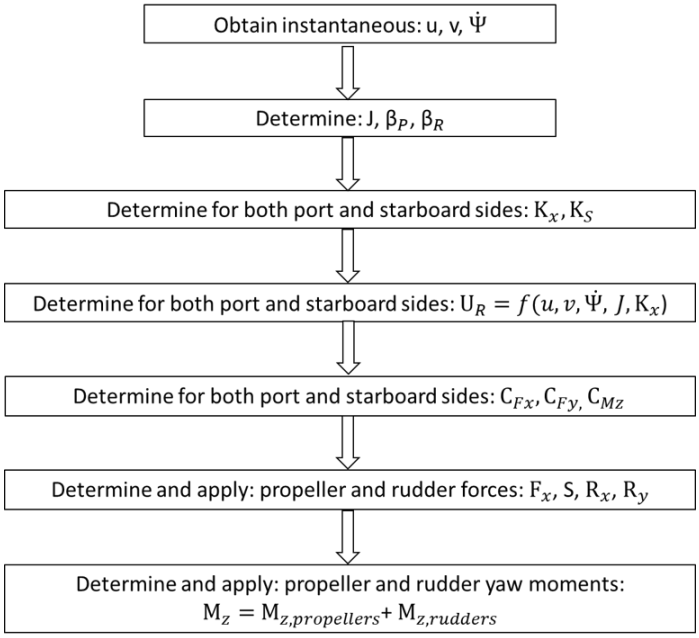

Figure 6.2: Flow chart for the implementation of the data-driven propeller and rudder model for the ONR Tumblehome maneuvering simulations.

motion and to the flow via the body force term in the Navier Stokes equations. The propeller forces applied for this case include the propeller axial force based on the force coefficient K_X and the propeller side force S based on the side force coefficient K_S. The yaw moment of the propellers are included by accounting for the side force of each propeller and the distance aft of the vessel center of gravity, along with the moment engendered due to the discrepancy in the thrust between the port and starboard side propellers times their distance off centerline of the vessel. The rudder force coefficients C_{Fx}, C_{Fy}, and C_{Mz} are used to determine the resistance, side force, and yaw moment of the rudder directly using Eqn. 2.15 to Eqn. 2.17. A flow chart showing the process of implementing the data-driven propeller and rudder model for this vessel is shown by Fig. 6.2.

6.2 Calm Water Turning Circle

Using the regression based data-driven model the maneuvering characteristics of the vessel are evaluated at a Froude number of 0.2 and compared to experimental results (*SIMMAN* (2020)). As discussed earlier, there is uncertainty in the data-driven model due to CFD discretization error, turbulence modeling error, and modeling error in the data-driven model itself. Additionally, there is discretization error in the VOF model as well as uncertainty due to turbulence modeling. In this section the effect of the VOF grid density is examined first using a single data-driven propeller and rudder model. Additionally, the effect of using different data-driven models trained with the different training grids, number of sample points, and different turbulence models for training the model are examined. As part of this the effect of including the side force of the propellers is examined. Finally, the effect performing the VOF maneuvering simulation with the Spalart Allmaras turbulence model versus the $k-\omega$ SST turbulence model is examined.

6.2.1 Effect of VOF Grid Density

The first 30 seconds of the turning circle maneuver from the release point is examined with the series of five different VOF grids. This length of time is sufficient for the vessel to reach its maximum transfer such that the numerical quantities of the maneuver can be compared. The yaw rate, forward velocity made dimensionless by the initial forward velocity of u_o of 1.11 m/s, and sway velocity are shown in Fig.6.3-6.5 for the five different grids examined relative to the experiment. These simulations use the propeller model trained with the eight point LHS set of data with the G2 double-body grid using the Spalart Allmaras turbulence model.

The maximum advance predicted with all of the grids is within 2.5% of each other. Similarly, the maximum transfer is predicted withing 3.5% of each other. The steady state yaw rate and forward velocity is similar between all of the grids examined.

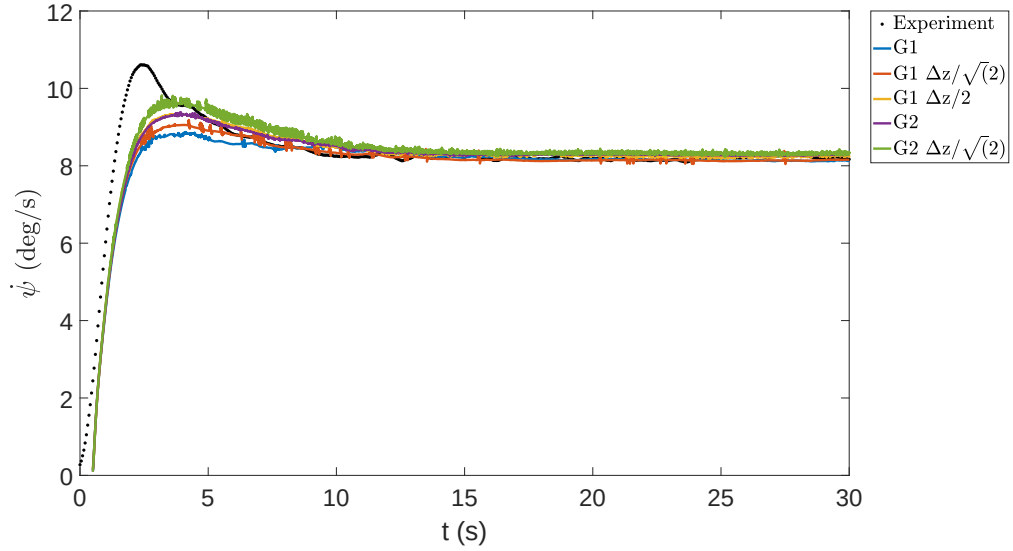

Figure 6.3: Yaw rate for ONR Tumblehome turning circle maneuver at F_n=0.2 and δ=-35° with eight point G2 Spalart Allmaras set of training data for the data-driven propeller and rudder model with different VOF grids relative to experimental results (*SIMMAN* (2020)).

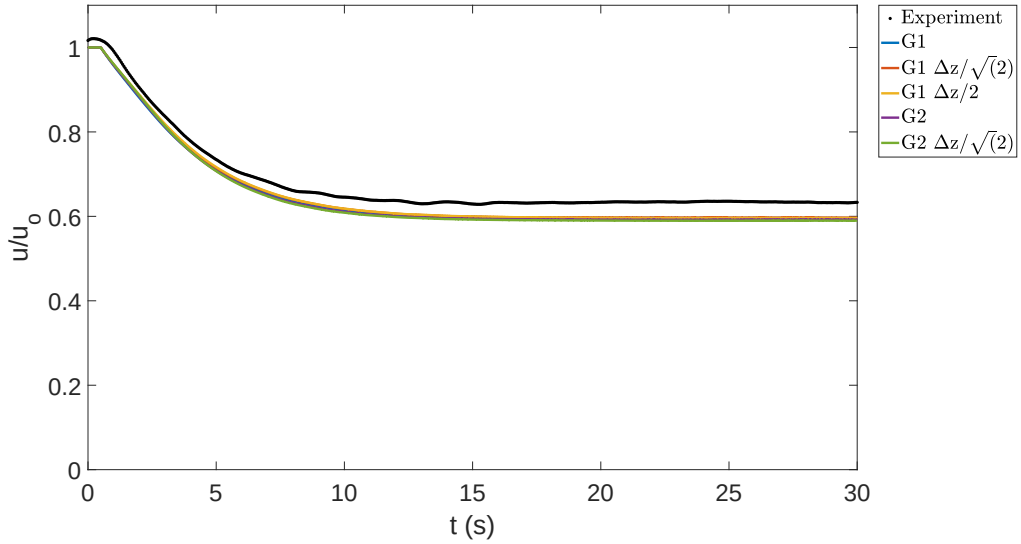

Figure 6.4: Forward velocity for ONR Tumblehome turning circle maneuver at F_n=0.2 and δ=-35° with eight point G2 Spalart Allmaras set of training data for the data-driven propeller and rudder model with different VOF grids relative to experimental results (*SIMMAN* (2020)).

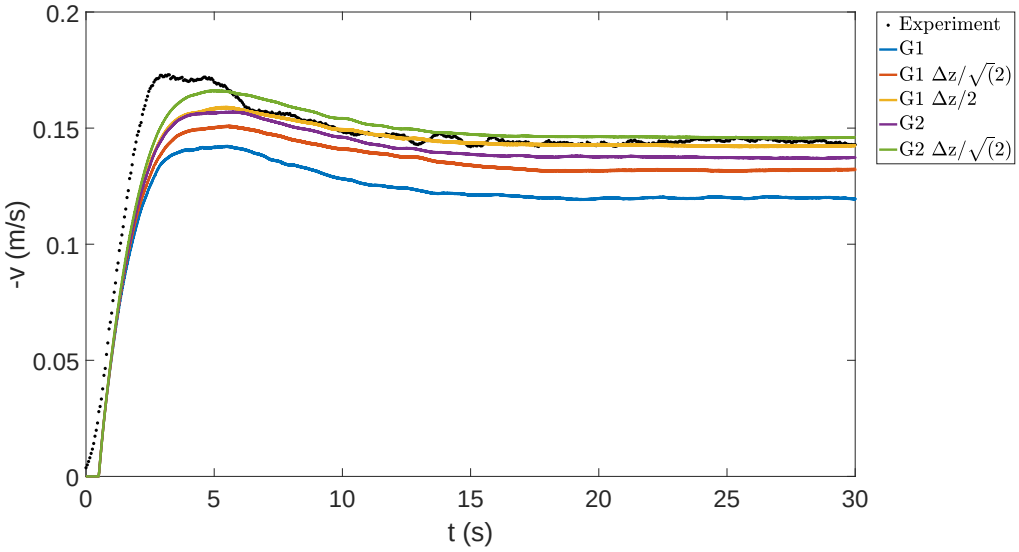

Figure 6.5: Sway velocity for ONR Tumblehome turning circle maneuver at F_n=0.2 and δ=-35° with eight point G2 Spalart Allmaras set of training data for the data-driven propeller and rudder model with different VOF grids relative to experimental results (*SIMMAN* (2020)).

The simulations predict a steady state forward velocity that is within 1.3% of each other; however, all of the cases under-predict the forward velocity of the experiment. Similarly all of the grids predict steady state yaw rates within 3% of each other. The most significant difference is between the prediction of the sway velocity. The largest difference is between the finest grid (G2 $\Delta z/\sqrt{2}$) and the coarsest grid (G1) for which there is over an 18% difference with respect to the finer grids value. This difference is likely caused by the refinement of the waterline. As the waterline is better refined there is better agreement with the finer grids and with the experiment. Since refining the waterline is rather inexpensive in comparison to refining the whole domain, the G1 $\Delta z/2$ grid is used for the rest of this study.

6.2.2 Effect of Different Body Force Models on Maneuver

Six different regression based data-driven propeller and rudder models are examined using the G1 $\Delta z/2$ VOF grid. The models include: the model trained with the

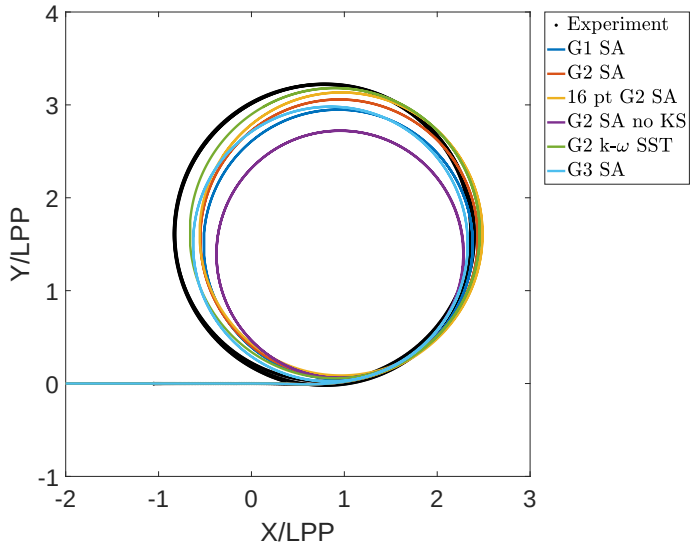

Figure 6.6: Trajectory of ONR Tumblehome turning circle maneuver at F_n=0.2 and δ=-35° with G1 $\Delta z/2$ VOF grid with different training sets for the data-driven propeller and rudder model relative to experimental results (*SIMMAN* (2020)).

G1 double body grid with the Spalart Allmaras turbulence model with eight LHS points (G1 SA), the model trained with the G2 double body grid with the Spalart Allmaras turbulence model with eight LHS points (G2 SA), the G2 SA model with no propeller side force (G2 SA no KS), the model trained with the G2 double body grid with the Spalart Allmaras turbulence model with sixteen LHS points (16 pt G2 SA), the model trained with the G2 double body grid with the $k-\omega$ SST turbulence model with eight LHS points (G2 $k-\omega$ SST), and the model trained with the G3 double body grid with the Spalart Allmaras turbulence model with eight LHS points (G3 SA). Figs. 6.6-6.9 shows the trajectory of the turning circle maneuver as viewed from above, the yaw rate as a function of time, the dimensionless forward velocity as a function of time, and the sway velocity as a function of time for the different models examined.

The results that agree best with the experiment (*SIMMAN* (2020) are those generated using the G2 $k-\omega$ SST data-driven model, while the worst results with respect

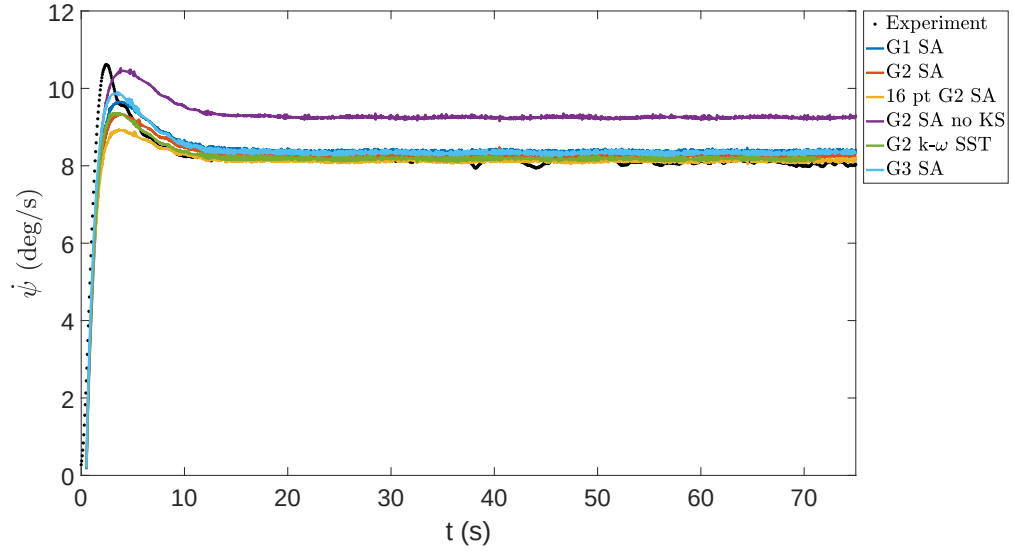

Figure 6.7: Yaw rate for ONR Tumblehome turning circle maneuver at F_n=0.2 and δ=-35° with G1 $\Delta z/2$ VOF grid with different training sets for the data-driven propeller and rudder model relative to experimental results (*SIMMAN* (2020)).

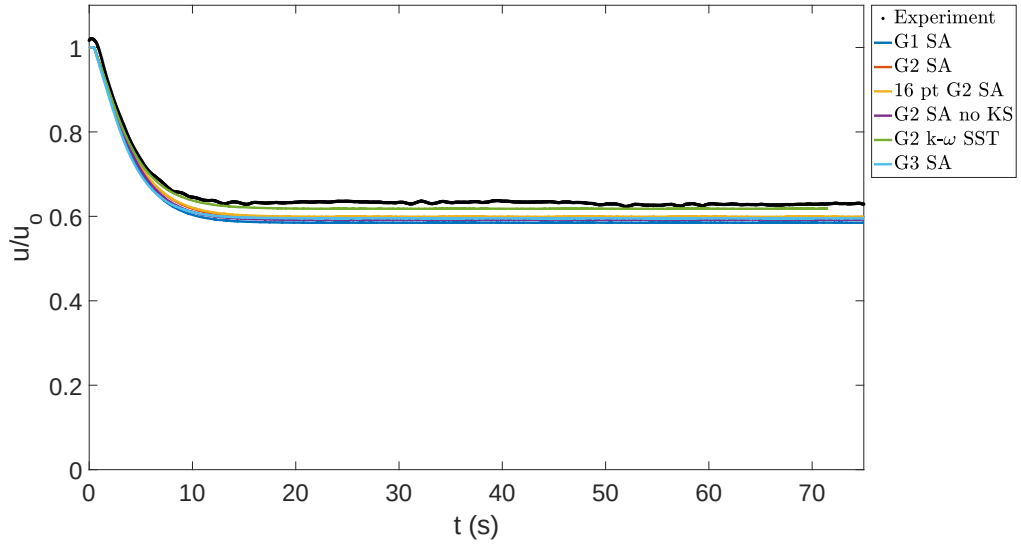

Figure 6.8: Forward velocity for ONR Tumblehome turning circle maneuver at F_n=0.2 and δ=-35° with G1 $\Delta z/2$ VOF grid with different training sets for the data-driven propeller and rudder model relative to experimental results (*SIMMAN* (2020)).

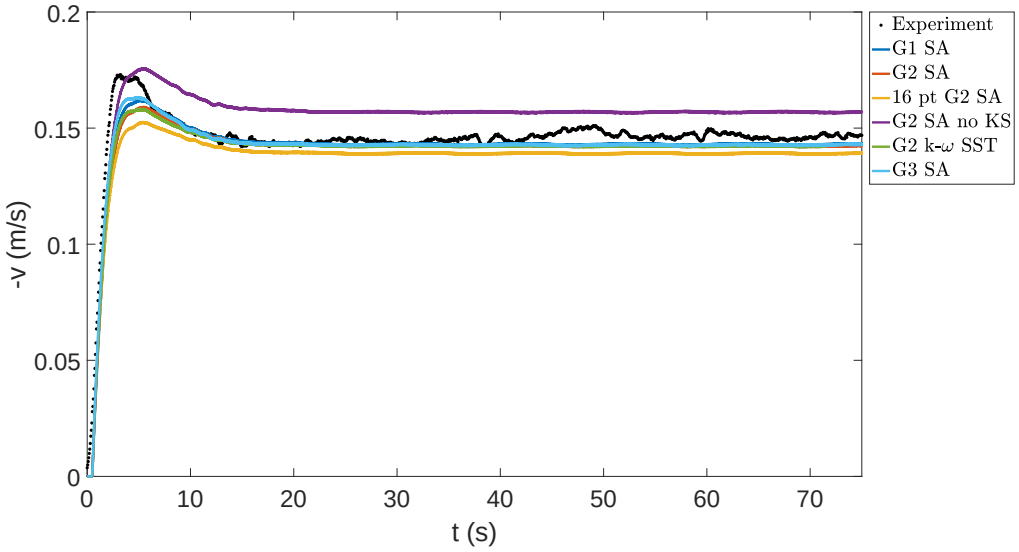

Figure 6.9: Sway velocity for ONR Tumblehome turning circle maneuver at F_n=0.2 and δ=-35° with G1 $\Delta z/2$ VOF grid with different training sets for the data-driven propeller and rudder model relative to experimental results (*SIMMAN* (2020)).

to the experiments is the case that neglects the propeller side force. This illustrates the importance of considering the propeller side force. The propeller side force opposes the side force of the rudder, which reduces the magnitude of the yaw rate and the sway velocity in the turn as illustrated by Fig. 6.7 and Fig. 6.9. Thus, when the side force of the propellers is neglected, the turning circle is too tight as shown by Fig. 6.6.

All of the cases examined under-predict the forward speed calculated from the experiment. The case that uses the G1 SA model has the worst agreement with the experimental forward speed, with less than 7% error; whereas the G2 $k-\omega$ SST model has the best agreement with the experiment for which the error is less than 2%. The most substantive difference in terms of calculating forward speed is what turbulence model is used train the data-driven model. There is less than 1% difference in the prediction of forward speed if 8 samples are used versus 16 samples to train the model with the G2 SA model. The steady state forward velocity using the G2 SA model

and the G3 SA model are nearly identical, while the G1 SA predicts approximately 2% less forward speed than either of the models trained with the finer grids.

All of the cases examined over-predict the steady turning yaw rate of the experimental results. In the case that neglects the propeller side force the yaw rate is over-predicted by nearly 15%. On the other hand, all of the cases that include the propeller side force predict the yaw rate to within 5% during the steady turn.

The G2 $k-\omega$ SST model over predicts the steady yaw rate by less than 1%. This slight over-prediction of yaw rate, coupled with the small under prediction of forward velocity and sway velocity leads to a turning circle that is tighter than the experiment, but in the best agreement with the experiment of all cases examined here. The turning circle characteristics predicted with the G2 $k-\omega$ SST model VOF simulation are within 5% of reported values based on *Sanada et al.* (2019) who reported turning +/- 35° rudder turning circles at this Froude number. Since the vessel is symmetric the discrepencies between the port and starboard turn in *Sanada et al.* (2019) have been averaged and the advance is predicted within 5%, the transfer is predicted within 3%, the tactical diameter is predicted within 1%, and the turning radius is within 2.5% if the G2 $k-\omega$ SST model is used for the VOF simulation.

Linear Regression Coefficients The linear regression coefficients for four different sets of training data are shown in Tables B.11-B.14.

6.2.3 Computational Cost Discussion

The computational cost of both the training simulations for the data-driven model as well as the maneuvering simulation cost are important to consider. The computational cost of the maneuver is significantly reduced when the data-driven propeller and rudder model is used. The time-step size is approximately the time it takes the propeller to rotate seventeen degrees. The cost to complete a single turn of 360° is

only 732 core hours (629 core hours once the rudder is actuated and 103 core hours to accelerate the vessel from rest) on the G1 $\Delta z/2$ VOF grid, for which the simulation length is approximately 55 s in model time. The training costs associated with the double-body grid CFD training simulations are: 1,190 core hours for the G1 grid, 4,344 core hours for the G2 grid, and 18,858 core hours for the G3 grid. The sample point correlating to LHS point seven is used to generate the reference training cost. To note, all cases have a limit of a time-step to not exceed 1° of propeller rotation per time-step, but also use a Courant limiter which drives up the computational cost for the G3 grid since this case has a time-step of less than 1° per time-step. Also, 40 cores are used for the G1 grid, 120 cores are used for the G2 grid, and 200 cores are used for the G3 grid, thus, some non-linear parallelization effects drive up the computational cost for the G3 grid, since the computational efficiency of decomposing the domain is less when additional cores are used.

In contrast, maneuvers that use a discretized propeller with an overset grid method are much more expensive. *Wang and Wan* (2018) performed a turning circle maneuver of the ONR Tumblehome in waves with a δ of 35° and a F_n of 0.2. This study required 48,000 core hours to model 78 seconds of model time on a 7.13 million cell grid. Similarly, *Wang et al.* (2018) showed that a zig-zag maneuver required nearly 13,000 core hours to model the maneuver in calm water and 14,000 core hours to model the maneuver in waves to simulate almost 20 s of model time.

The use of the propeller and rudder model allows for high fidelity simulation with small temporal and spatial discretization on the propeller and rudder for training the model (G2 or G3), while the VOF maneuvering simulation can be performed on a much coarser grid (G1). It is important to consider both the training cost as well as the cost when the maneuver is performed. It is found in the results that using only eight points for training the model is sufficient to generate a model with sufficient accuracy, so the model can be trained and implemented in a maneuvering simulation

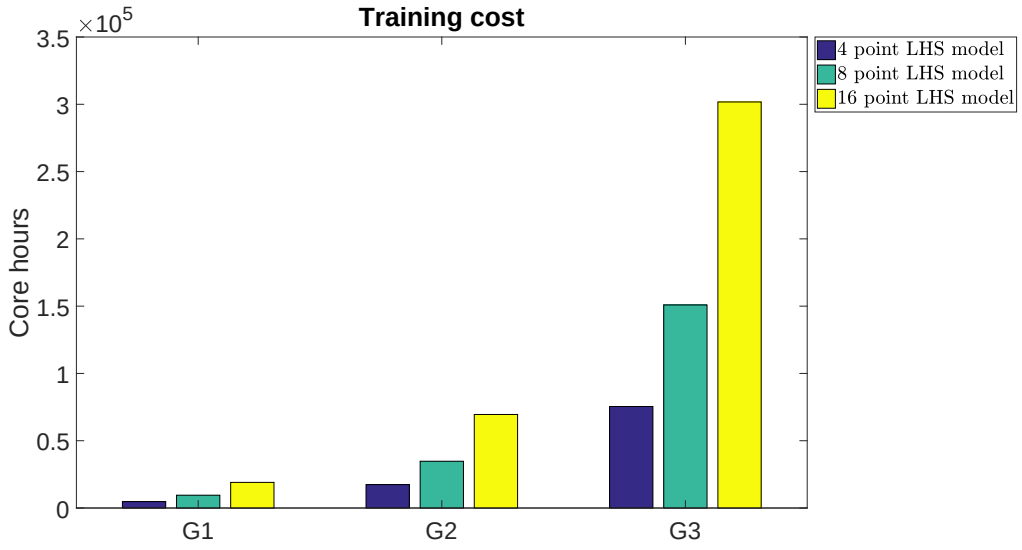

Figure 6.10: Computational cost to train different sets of training data for the ONR Tumblehome.

at less than the cost of performing a maneuvering CFD simulation with a discretized propeller and rudder. For reference, each double-body simulation used to train the model is run for 5.56 s in model time, so the total length of model time for generating eight training points is 44.6 s.

Fig. 6.10 shows the training cost for the G1, G2, and G3 models with different sets of training data, for which the sample point correlating to LHS point seven is used to generate the reference training cost. Fig. 6.11 shows the cost of training a model using eight samples of double-body CFD simulation on the G1, G2, and G3 grids and implementing it with a VOF grid in comparison to using a discretized rotating propeller. For each color the solid line represents the cost using a double-body discretized propeller simulation while the implemented cost of the data-driven model trained with the eight point set for each respective double-body grid is shown with the dashed lines. This figure again highlights that the computational cost of the implemented model is dominated by the training cost for all grids examined like was shown with the KCS analysis. Also, the cost to model a maneuvering ship is

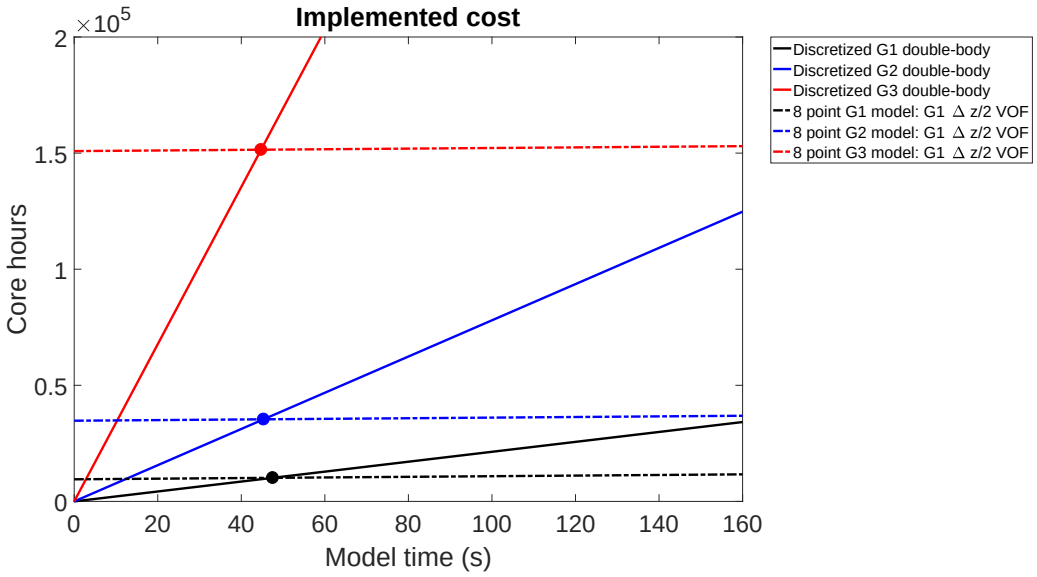

Figure 6.11: Computational cost to train and implement a data-driven model trained with eight points on the different double-body grids and implemented on the G1 $\Delta z/2$ VOF grid compared to using discretized propellers for the ONR Tumblehome.

significantly less expensive than modeling a discretized propeller. The dots show the cross-over point at which point it is less expensive to train the data-driven model and use it to model the maneuvering vessel with the G1 $\Delta z/2$ VOF grid. It is less expensive to train and implement the model for all of the different cases than use a discretized propeller even if only one turning circle is examined. Furthermore, it is illustrated that as finer training grids are used, the cross-over point moves to the left, thus as finer grids are used to train the model, they can be implemented on a coarser VOF grid for maneuvering simulations which reduces cost. If multiple simulations are performed, the computational cost of the maneuver is small relative to the training cost. Thus, if it is desired to perform multiple maneuvering simulations, this method is especially appropriate. This method enables the computationally efficient analysis of analyzing the maneuvering characteristics of the vessel in waves and at multiple Froude numbers.

6.3 Turning Circle in Waves

The vessel is analyzed in regular waves and compared to the *SIMMAN* (2020) experiments. The Froude number is 0.2, the wave height is 0.063 m at model scale, the wave period is 1.42 s, and the wave length is equal to the ship length. The vessel is in head seas when the rudder is actuated to -35°, such that the waves propagate in the $-X$ direction.

The experimental result are compared to three CFD approaches. The eight point Spalart Allmaras turbulence model set of data is used to train the propeller and rudder regression based data-driven model for two of the cases. One of these cases uses the G1 VOF grid with the waterline refined by a factor of $\sqrt{2}$, while the other has the waterline refined by a factor of two; these are denoted by `G1/`$\sqrt{2}$ `SA` and `G1/2 SA` respectively. The eight point $k-\omega$ SST turbulence model set of data is used to train the propeller and rudder linear regression based model. The propeller and rudder model is used on the G1 grid with the waterline refined by a factor of two and this case is denoted `G1/2` $k-\omega$ `SST`. All of the CFD cases use the Spalart Allmaras turbulence model for the VOF simulations. The CFD simulations are run for a total of 68 seconds once the rudder is actuated and the vessel is released.

Fig. 6.12 shows the trajectory of the experimental results and the CFD predictions. Arrows are also included for each case which shows the effect of the second order drift of the vessel turning in waves. The arrows connect the point in the trajectory for when the ψ is equal to 180° and 540°; the magnitude of this vector is the drift distance D_d, and the drift angle μ_d is the angle of this vector. μ_d and drift distance D_d are shown by Table 6.2. This illustrates the angle of μ_d with respect to the initial heading of the vessel as well as the magnitude of the total drift distance over the course of a turning circle. The CFD simulations are run for a duration of 68 seconds after the rudder is deflected, which is sufficient for the vessel to perform one and a half turning circles as shown by Fig. 6.13. The drift distance predicted by the different CFD

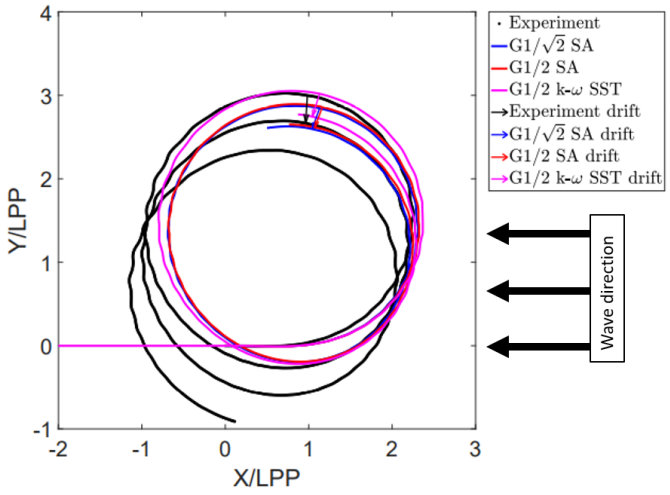

Figure 6.12: Trajectory for ONR Tumblehome turning circle maneuver at F_n=0.2 and δ=-35° in waves relative to experimental results (*SIMMAN* (2020)).

Table 6.2: Turning circle maneuvering characteristics for ONR Tumblehome in waves.

Case	μ_d (degrees)	D_d/L_{WL}	Tactical Diameter/L_{WL}
Experiment	-93.3	0.335	3.00
G1/$\sqrt{2}$ SA	-109.6	0.263	2.84
G1/2 SA	-110.5	0.260	2.86
G1/2 $k-\omega$ SST	-107.7	0.287	3.02

techniques are within 10% of each other. The G1/2 $k-\omega$ SST case agrees best with the experiment in terms of the overall trajectory, however, even with this case the drift distance is 14% less than the experiment. On the other hand, this case predicts the tactical diameter of the first turn to within one percent of the experiment; conversely the G1/$\sqrt{2}$ SA has over five percent error with respect to the experimental tactical diameter. The μ_d predicted with the CFD causes the vessel to drift more in the $-X$ direction (in the same direction as the waves). On the other hand, the experiment predicts a magnitude of μ_d that is approximately 16° less than the CFD predicts, such that the drift angle is more aligned with the $-Y$ axis (or towards starboard from the initial vessel heading).

Overall, the agreement between the G1/$\sqrt{2}$ SA and G1/2 SA cases is very good,

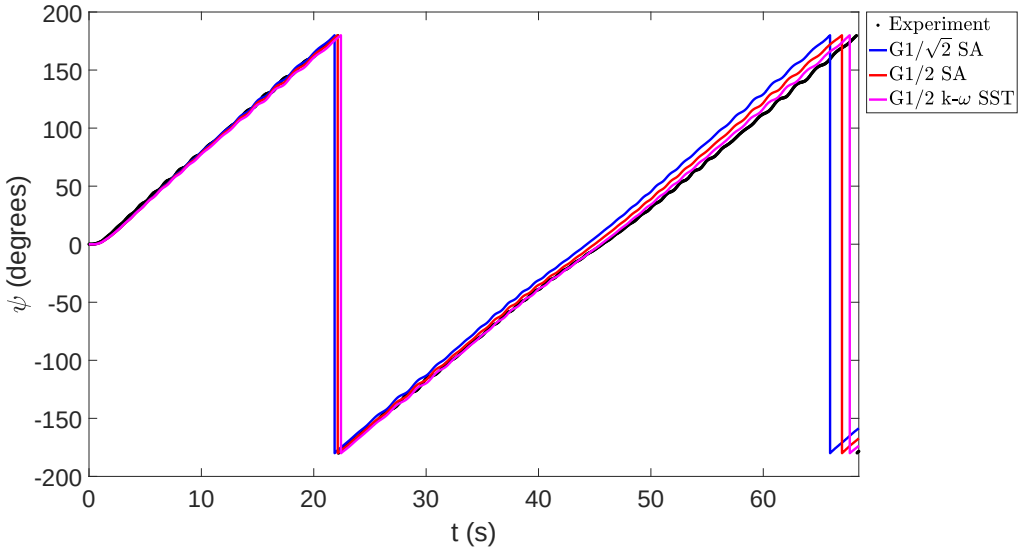

Figure 6.13: ψ for ONR Tumblehome turning circle maneuver at F_n=0.2 and δ=-35° in waves relative to experimental results (*SIMMAN* (2020)).

similar to the calm water case in terms of the trajectory as well as forward speed and sway velocity; whereas the G1/2 $k-\omega$ SST case agrees better with the experiment compared to the other CFD cases. Fig. 6.14 shows the forward velocity and Fig. 6.15 shows the sway velocity for the different cases as a function of time after the rudder is actuated. Similar to the calm water turning circle, the propeller and rudder model trained with data using the $k-\omega$ SST turbulence model leads to a slightly larger u compared to the cases that use the Spalart Allmaras turbulence model for training. All of the CFD simulation predictions track relatively well with the experimental u, but a phase shift and oscillation amplitude of u is observable. Similarly, the CFD predictions track well with the experimental v, but the oscillation amplitude of v about its mean is larger in the experiment.

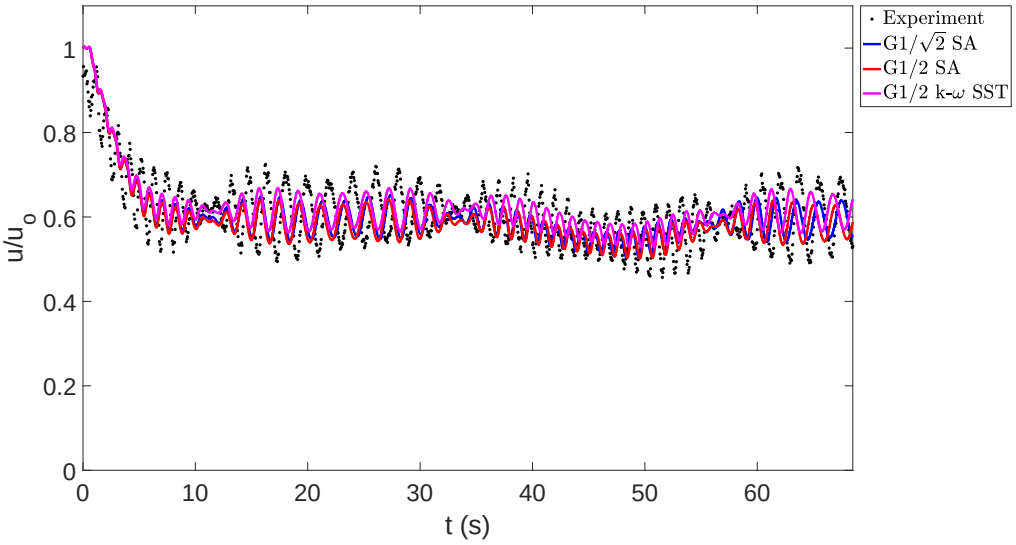

Figure 6.14: Forward velocity for ONR Tumblehome turning circle maneuver at F_n=0.2 and δ=-35° in waves relative to experimental results (*SIMMAN* (2020)).

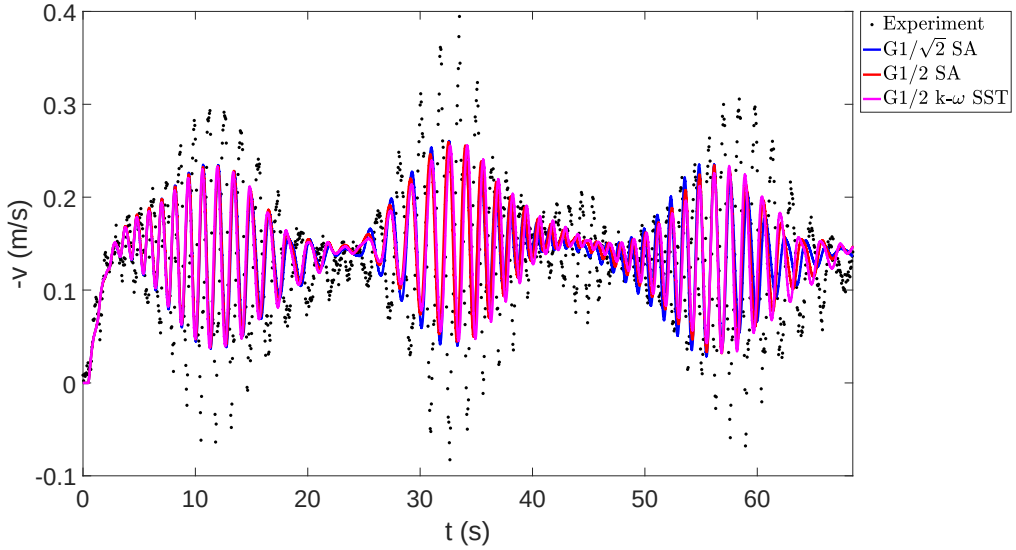

Figure 6.15: Sway velocity for ONR Tumblehome turning circle maneuver at F_n=0.2 and δ=-35° in waves relative to experimental results (*SIMMAN* (2020)).

6.4 Froude Number Effects on Maneuver

To further evaluate the data-driven propeller and rudder model, the turning circle parameters at different Froude numbers can be evaluated. The propeller model is trained at the equivalent speed and propeller revolution rate expected for a turning circle maneuver at a Froude number of 0.2. By evaluating the data-driven model for maneuvers at both higher and lower speed it can illustrate how well the model scales and the potential versatility of the model. *Sanada et al.* (2019) present the tabulated data of the advance, transfer, tactical diameter, and turning radius for the ONR Tumblehome performing a turning circle maneuver with the rudders deflected to +/- 35 ° at Froude numbers of 0.1, 0.2, and 0.3. *Sanada et al.* (2019) present slight differences between the port and starboard turns. In the CFD model, since it symmetric regardless of weather it is a port or starboard turn, the results of *Sanada et al.* (2019) have been averaged and presented in Fig. 6.16. Fig. 6.16 compares these experimental results for each Froude number relative to the CFD prediction. The CFD prediction is made using the G1 VOF grid with the waterline refined by a factor of two (G1 $\Delta z/2$) with the propeller model trained with the eight point G2-$k-\omega$ SST set of data.

The advance, transfer, tactical diameter, and radius are calculated within 5% of the average experimental results for each Froude number examined. The Froude number of 0.3 contains the largest error relative to the experiments for the transfer, tactical diameter, and radius predictions, for which it under-predicts the transfer by 4.62%, the tactical diameter by 4.56%, and the radius by 4.86% relative to the average of the experimental results. As the Froude number increases, the effect of the wave resistance increases whereas for lower Froude numbers the resistance is dominated by friction; thus, a reason for more significant error in the Froude number of 0.3 case could be that the free surface is not refined sufficiently. The most significant error in the calculation of the advance is the Froude number of 0.2 case which over-predicts

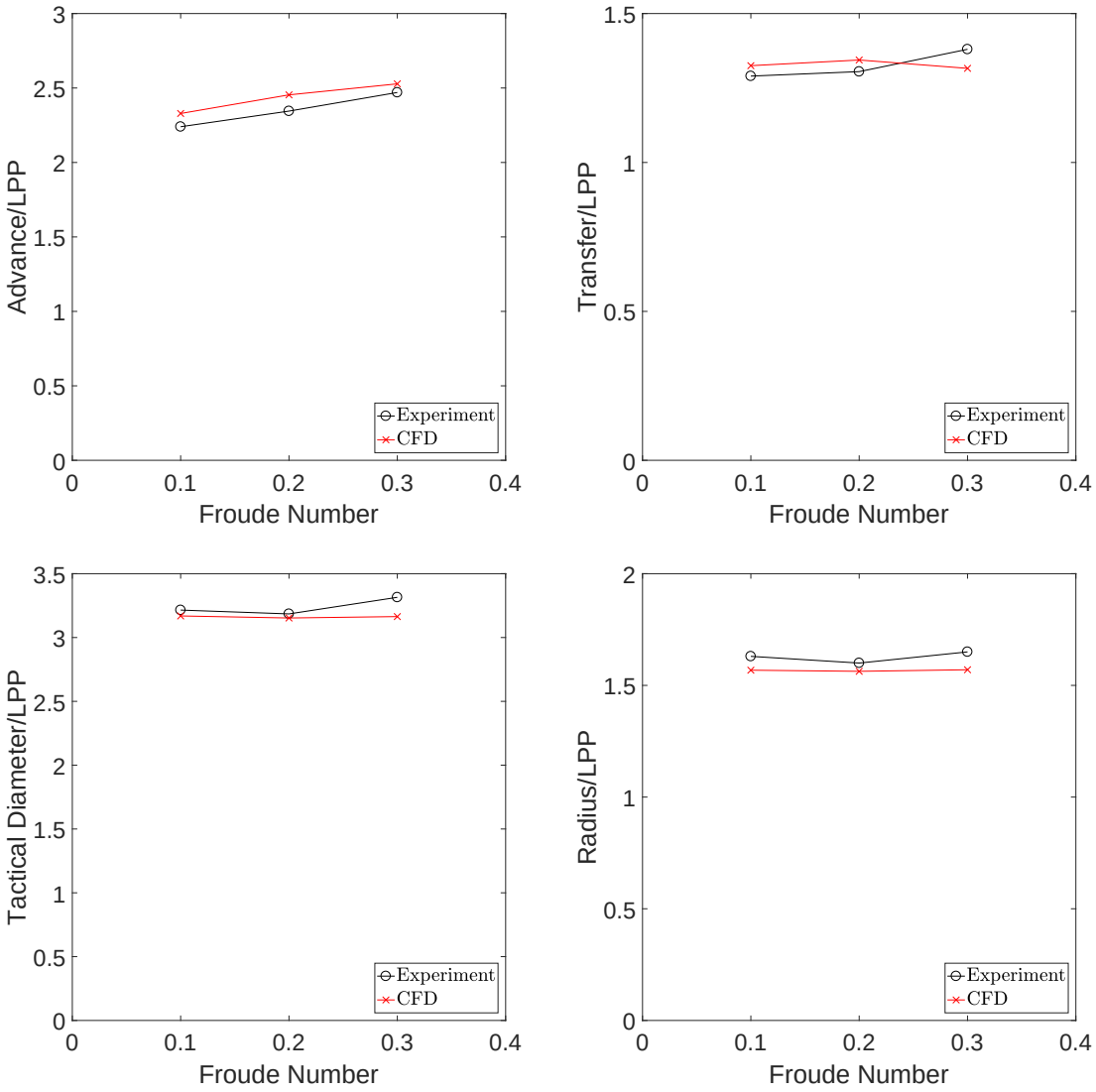

Figure 6.16: Comparison between *Sanada et al.* (2019) experimental results and CFD for turning circle parameters of ONR Tumblehome with different Froude Numbers. As a function of Froude number, the advance is shown in the upper left, the transfer is shown in the upper right, the tactical diameter is shown in the lower left, and the radius is shown in the lower right.

the advance by 4.63%. The results are promising since it indicates that if training is performed at a given speed and propeller revolution rate, the same model is applicable to maneuvers at different speeds than it was trained with.

CHAPTER VII

Conclusion

This work demonstrates how data-driven modeling and machine learning techniques can be used to determine the propeller and rudder forces for a maneuvering vessel. In this work, the data-driven model for the propeller(s) and rudder(s) has been trained with RANS CFD calculations using a discretized rotating propeller with a discretized deflected rudder. The operating space of the vessel is analyzed in terms of the advance coefficient and the oblique flow angle to develop data-driven models that can be easily scaled to maneuvers at alternative propeller revolution rates.

This work has investigated the effect of the quantity and fidelity of training data on the accuracy of a data-driven propeller and rudder model. Specifically, different levels of grid refinement, different turbulence models, and different data-driven modeling techniques including linear regression and Gaussian process regression have been examined in detail. This study illustrates how the fidelity of the training data propagates to the accuracy of the data-driven model and also illustrates the effect of how sparsely sampled the operating dimensionless space can be while still obtaining accurate results. The results demonstrate a substantial reduction in computational cost, that justify using the approach of using CFD calculations to train a data-driven propeller and rudder model even if just one maneuver is performed; but the method is applicable in scenarios when multiple maneuvering simulations are desired as long

as the rudder angle is consistent with the training rudder angle. Some examples of this include maneuvering in waves, maneuvering with different initial forward speeds, or the analysis of the effect of displacement on a maneuver.

This study has shown that with linear regression and only eight training points, the maneuvering characteristics of a vessel can be predicted with accuracy of within 5% of experiments. When using a data-driven propeller and rudder model, the computational cost of the maneuvering simulation is dramatically reduced from a simulation which uses a discretized rotating propeller. Additionally, by removing the need to discretize and actuate the rudder, the problem setup is simplified and the need for certain moving mesh treatments like overset grids or immersed boundary methods is removed. The data-driven model is trained with a CFD simulation with a fine spatial and temporal time-step to capture the complex flow around the propeller and rudder, but in maneuvering simulations, the data-driven model can be implemented on a VOF grid that is spatially coarser and takes a significantly larger time-step.

Data-driven methods can only be expected to determine output parameters as accurately as the model that they are trained with. Depending upon the fidelity of data used to train and validate the data-driven model, different approaches can be used. For training and validation data that inherently contains discretization and/or turbulence modeling error, linear regression is a suitable data-driven approach. In this work, it as been demonstrated that the maneuvering characteristics of a vessel can be accurately predicted with a data-driven propeller and rudder model trained with only eight samples of the propeller and rudder operating in the behind condition. Alternatively, GPR could be used if σ_n is calculated from the error of the training data; otherwise, the model may over-fit the training data which contains uncertainties. If on the other hand, the data used to train the model is very high fidelity and the discretization error is small, GPR could be a very good choice but it would also require more effort to implement in a custom maneuvering CFD package. It has

been shown in this study that for a low-dimensional space like this problem in which only J and β are used, as more training samples are used the GPR prediction of the validation points improves. Thus, with sufficient training data, and high fidelity data a GPR model can very accurately predict the true propeller and rudder force of a vessel performing a maneuver, if the cost can be justified. If on the other hand, the cost of training a very high fidelity model is out of reach due to computational cost, quite reasonable predictions can be made training a data-driven propeller and rudder model with RANS CFD and linear regression. In the two case studies performed in this work, it has been demonstrated that a linear regression based data-driven propeller and rudder model trained with a moderately sized (on the order of six million cells) can be used to generate a model that leads to accurate maneuvering characteristics.

7.1 Future Work

In this work the focus is upon illustrating how a data-driven propeller and rudder model can be used to significantly reduce computational cost of a CFD based analysis of a maneuvering vessel. In this work only turning circle maneuvers were examined; however, this approach can be applied to other maneuvers like zig-zag maneuvers or course keeping maneuvers which would require varying rudder angles. In this work, the training for the data-driven propeller and rudder model was performed for a single rudder angle to illustrate how the forces of the propeller and rudder can be determined, the accuracy quantified, and how the model can be implemented in a CFD package. A third parameter can be included in the data-driven analysis to determine the forces based on varying rudder angles and the three dimensional parameter space (J, β, δ) could be samples with LHS. On the other hand, a series of common δ angles could be used to generate the models for slices through this three dimensional space in just the J-β plane. Depending upon the δ, the forces on the

propeller and rudder could be interpolated between different planes correlating to different values of δ. Additionally, the use of oblique flow angle in the data-driven model to predict the rudder force coefficients could be recast as the effective inflow angle to the rudder based upon the β_R as well as the rudder angle. This more general expression could allow for the analysis of other rudder angles without generating additional training data for a range of δ. Furthermore, the J-β space would likely change depending upon the value of δ, since for low values of δ, the vessel would not operate with as large of a β as it would for high values of δ. In this way, the training cost of a more generalized data-driven propeller and rudder model could be mitigated while maintaining accuracy.

While RANS CFD is capable of calculating viscous effects like separation, the fidelity of RANS CFD for heavily separated flows is not as good as higher fidelity methods like LES or DES. Higher fidelity calculations or experiments could be used to train the propeller and rudder model derived in this work and then implemented in a RANS VOF maneuvering simulation. With a large amount of very high fidelity propeller and rudder forces for the operating parameter space of the vessel, a high fidelity data-driven model can be constructed. For high fidelity data, with low expected noise and σ_n, it may be more desirable to construct a Gaussian process regression based propeller and rudder model than a linear regression model, if sufficient data can be generated. However, for many applications, linear regression should provide sufficient accuracy and is easier to implement in a custom CFD package.

Ultimately, the practitioner of the method discussed in this work should use their engineering judgment in determining what fidelity of training data to use in conjunction with this model. In future work, a multi-fidelity method could be used in conjunction with either linear regression or Gaussian process regression to construct a data-driven propeller and rudder model that is trained with very high fidelity results. For example, RANS CFD could be used to train the model when heavily separated

flows are not expected, while DES or LES could be used to generate training and validation data for when heavily separated flow is expected. This model could be used in conjunction with a RANS VOF simulation that could determine the propeller and rudder forces with the fidelity of DES or LES, but at a cost of nearly a bare hull simulation. With modern computers, a maneuvering simulation with a discretized propeller with RANS CFD is limited to a select few number of simulations; whereas this data-driven method could enable the examination of many more maneuvers both in calm water and in waves. To further reduce the computational cost, the data-driven model could also be constructed with lower fidelity methods like BEMT augmented with higher fidelity viscous CFD simulations. This method could also enable engineers to iterate with multiple designs to improve the performance of a vessel. Ultimately, this could lead to safer and more efficient vessels.

This work has focused upon CFD applications of the model, however, the framework for the data-driven propeller and rudder model can be extended to other applications. This work could be implemented on board autonomous vessels to predict the force of the propeller and rudder. Furthermore, in the case of implementation on autonomous vessels, the algorithm could be improved by using real-world data to further train the model.

APPENDICES

APPENDIX A

Supplementary Tables and Figures for KCS Analysis

In this appendix are supplementary tables and figures for the KCS analysis.

A.1 KCS Latin Hypercube Sample Points.

Table A.1: Values for each of the LHS points for KCS analysis in the four point LHS set.

LHS point	J	β (degrees)	u (m/s)	v (m/s)
1	0.462	39.38	0.505	0.414
2	0.288	28.12	0.314	0.168
3	0.638	16.88	0.696	0.211
4	0.812	5.62	0.887	0.087

Table A.2: Values for each of the LHS points for KCS analysis in the eight point LHS set.

LHS point	J	β (degrees)	u (m/s)	v (m/s)
5	0.681	25.31	0.744	0.352
6	0.594	2.81	0.648	0.032
7	0.244	30.94	0.266	0.160
8	0.419	42.19	0.457	0.414
9	0.769	36.56	0.839	0.623
10	0.506	14.06	0.553	0.138
11	0.331	19.69	0.362	0.129
12	0.856	8.44	0.935	0.139

Table A.3: Values for each of the LHS points for KCS analysis in the sixteen point LHS set.

LHS point	J	β (degrees)	u (m/s)	v (m/s)
13	0.441	4.22	0.481	0.035
14	0.309	37.97	0.338	0.264
15	0.834	32.34	0.911	0.577
16	0.703	26.72	0.768	0.386
17	0.747	12.66	0.816	0.183
18	0.572	18.28	0.624	0.206
19	0.397	21.09	0.433	0.167
20	0.222	9.84	0.242	0.042
21	0.266	43.59	0.290	0.276
22	0.528	35.16	0.577	0.406
23	0.484	29.53	0.529	0.300
24	0.659	7.03	0.720	0.089
25	0.616	1.41	0.672	0.017
26	0.878	40.78	0.959	0.827
27	0.353	15.47	0.386	0.107
28	0.791	23.91	0.863	0.383

Table A.4: KCS double-body CFD grid y^+ for LHS point 2.

	G1	G2	G3
Hull y^+	21.6	9.1	11.2
Propeller y^+	17.7	15.4	7.8
Rudder y^+	13.6	10.4	6.8

Table A.5: Discretization error between different grids for $10K_Q$.

LHS point	G1 ($10K_Q$)	G2 ($10K_Q$)	G3 ($10K_Q$)	ϵ_{23} (%)	ϵ_{13} (%)
1	0.530	0.539	0.531	-1.369	0.266
2	0.606	0.633	0.625	-1.260	3.057
3	0.481	0.465	0.453	-2.596	-6.033
4	0.445	0.420	0.424	0.964	-5.033
5	0.448	0.436	0.417	-4.666	-7.390
6	0.498	0.506	0.512	1.036	2.674
7	0.637	0.653	0.647	-0.866	1.578
8	0.545	0.555	0.548	-1.283	0.553
9	0.382	0.376	0.351	-7.092	-8.869
10	0.540	0.523	0.521	-0.328	-3.649
11	0.621	0.616	0.608	-1.365	-2.162
12	0.428	0.394	0.380	-3.588	-12.598

A.2 Supplementary Tables for Data-Driven Model Development

This section contains supplementary tables that are useful for describing the data-driven model development for the KCS. Specifically, supplementary tables are shown to augment the discussion on the underlying discretization error and turbulence modeling error in the training data.

A.2.1 Discretization Error: Supplementary Tables

Table A.6: Discretization error between different grids for C_{Fx}.

LHS point	G1 (C_{Fx})	G2 (C_{Fx})	G3 (C_{Fx})	ϵ_{23} (%)	ϵ_{13} (%)
1	-0.259	-0.267	-0.274	2.446	5.314
2	-0.224	-0.233	-0.229	-1.724	1.839
3	-0.294	-0.320	-0.346	7.546	15.085
4	-0.304	-0.325	-0.339	4.053	10.252
5	-0.291	-0.305	-0.326	6.553	10.975
6	-0.312	-0.326	-0.334	2.420	6.649
7	-0.217	-0.216	-0.207	-4.331	-4.649
8	-0.242	-0.250	-0.253	1.134	4.294
9	-0.194	-0.213	-0.221	3.600	12.251
10	-0.274	-0.310	-0.315	1.367	13.066
11	-0.249	-0.258	-0.249	-3.478	-0.020
12	-0.287	-0.314	-0.341	7.802	15.784

A.2.2 Turbulence Modeling Error: Supplementary Tables

Table A.7: Turbulence modeling error between Spalart Allmaras and $k-\omega$ SST for $10K_Q$.

Grid	LHS point	Spalart Allmaras ($10K_Q$)	$k-\omega$ SST ($10K_Q$)	ϵ_T (%)
G1	1	0.530	0.526	0.708
G2	1	0.539	0.537	0.212
G3	1	0.531	0.539	-1.377
G1	2	0.606	0.623	-2.704
G2	2	0.633	0.643	-1.504
G3	2	0.625	0.645	-3.132
G1	3	0.481	0.514	-6.993
G2	3	0.465	0.522	-12.153
G3	3	0.453	0.532	-17.365
G1	4	0.445	0.422	5.268
G2	4	0.420	0.440	-4.929
G3	4	0.424	0.451	-6.336

Table A.8: Turbulence modeling error between Spalart Allmaras and $k-\omega$ SST for C_{Fx}.

Grid	LHS point	Spalart Allmaras (C_{Fx})	$k-\omega$ SST (C_{Fx})	ϵ_T (%)
G1	1	-0.259	-0.189	27.012
G2	1	-0.267	-0.185	30.761
G3	1	-0.274	-0.181	33.897
G1	2	-0.224	-0.213	5.089
G2	2	-0.233	-0.230	1.036
G3	2	-0.229	-0.216	5.410
G1	3	-0.294	-0.270	8.083
G2	3	-0.320	-0.261	18.390
G3	3	-0.346	-0.270	21.995
G1	4	-0.304	-0.322	-5.748
G2	4	-0.325	-0.281	13.594
G3	4	-0.339	-0.266	21.402

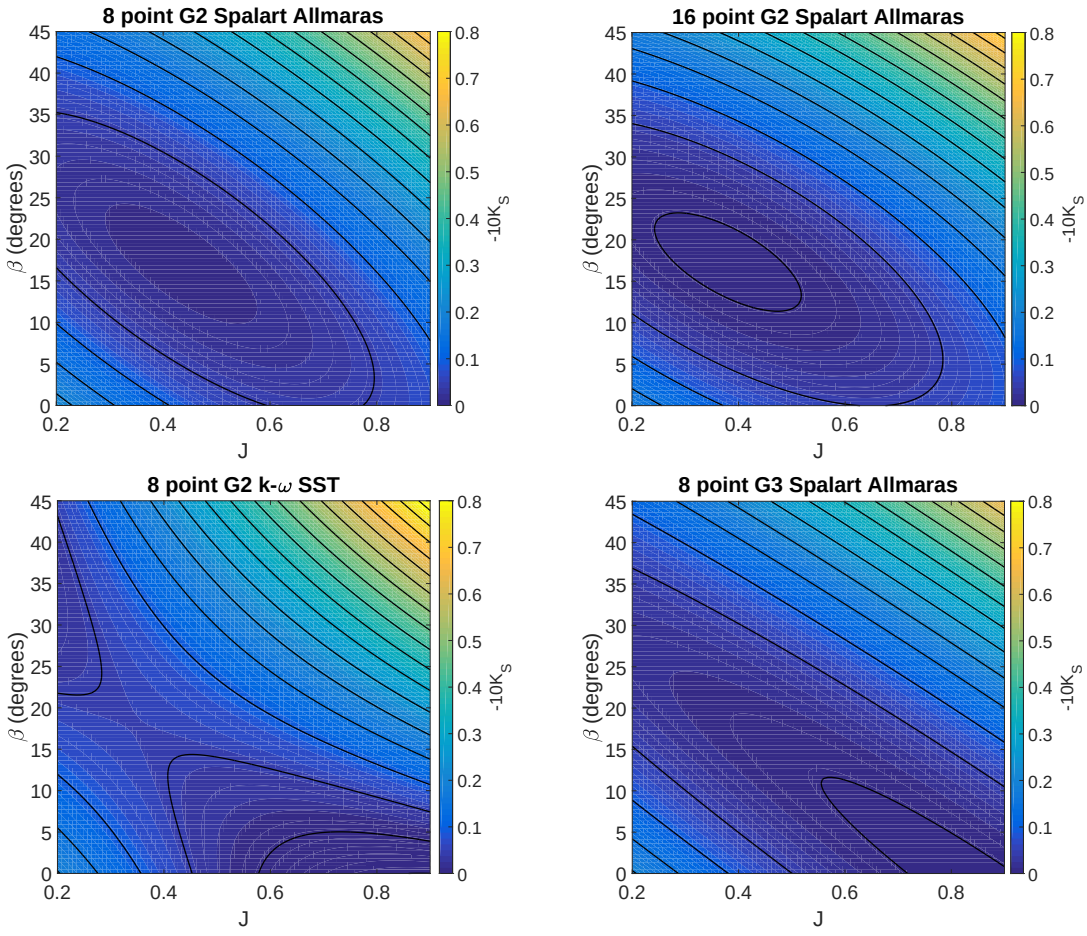

Figure A.1: Contours of -10K_S in J-β space for linear regression models trained with different data sets.

A.2.3 Comparison Between Linear Regression Models: Supplementary Figures

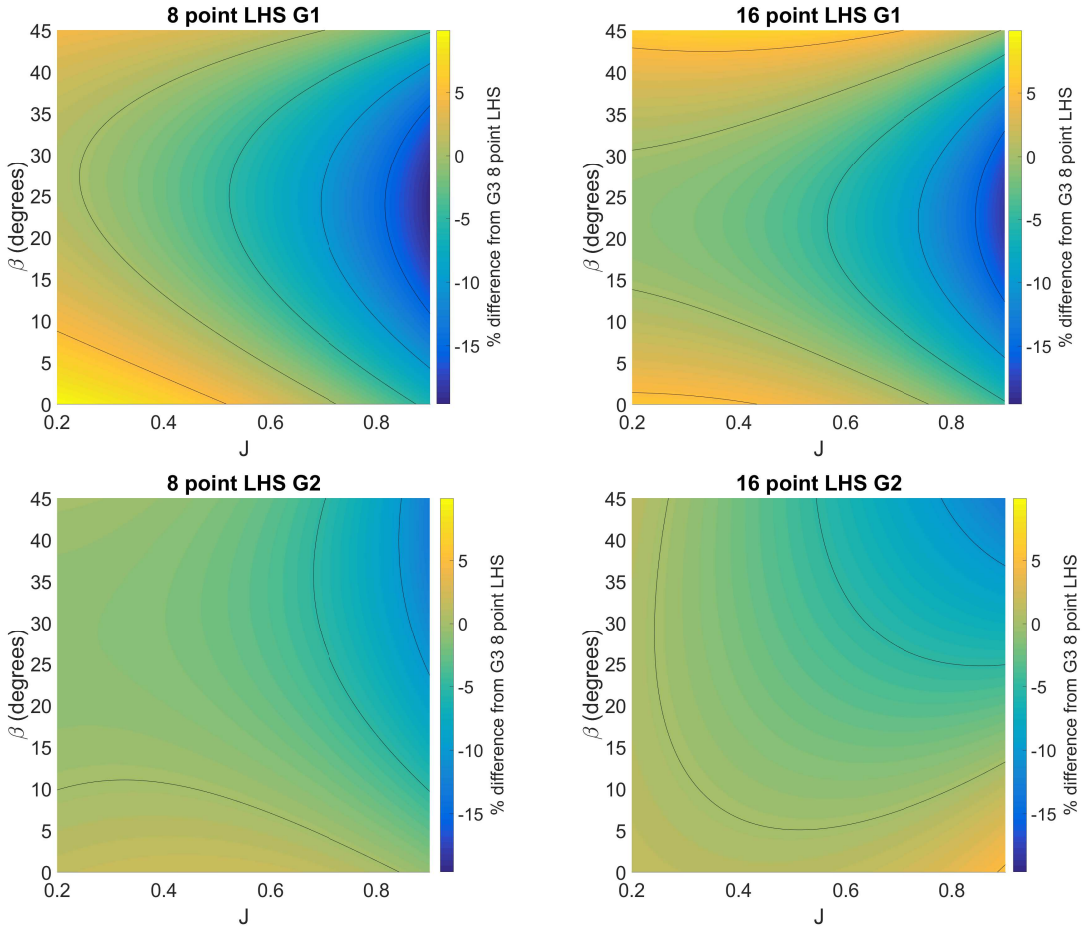

Figure A.2: Contours of percent difference of different linear regression models' prediction of $10K_Q$ with respect to the 8 point LHS G3 model in J-β space.

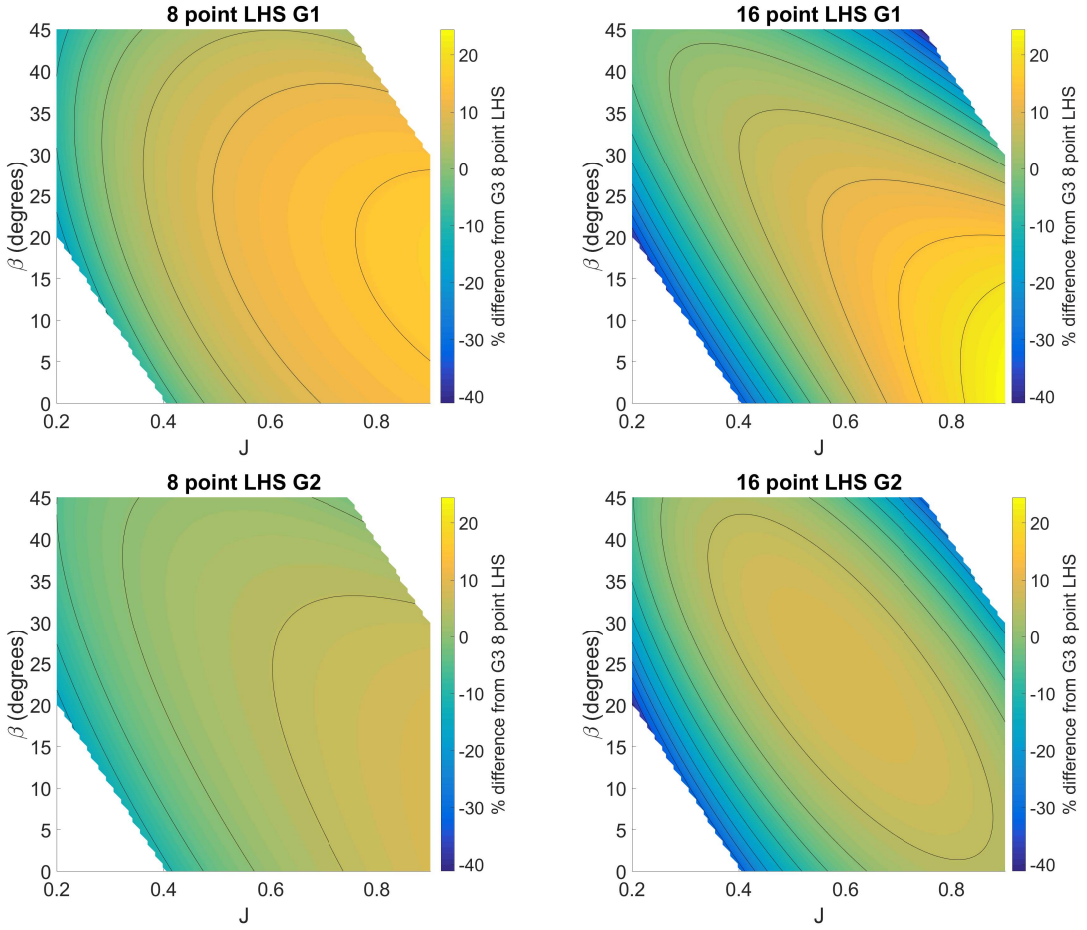

Figure A.3: Contours of percent difference of different linear regression models' prediction of C_{Fx} with respect to the 8 point LHS G3 model in J-β space.

A.3 Gaussian Process Regression Supplementary Figures.

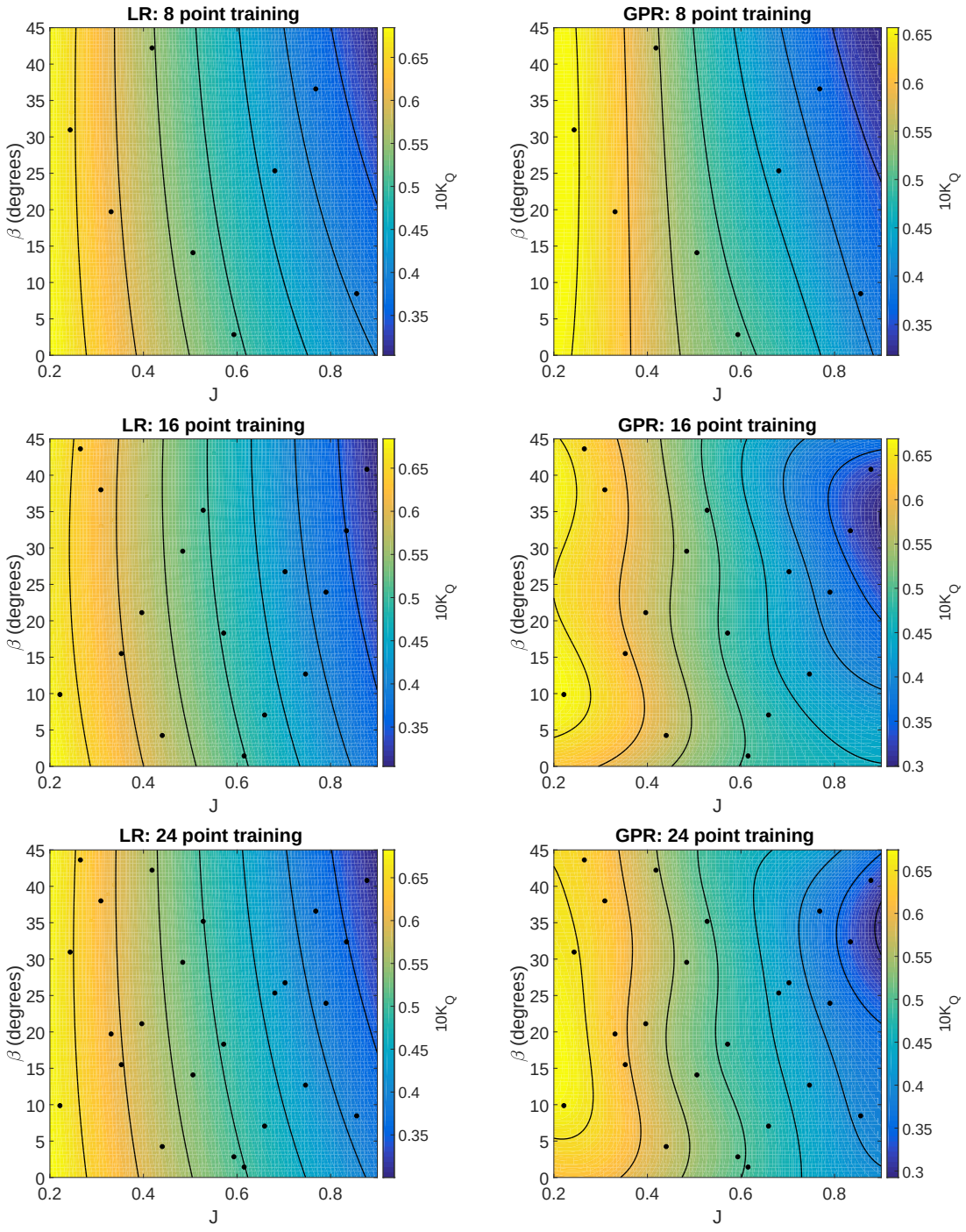

Figure A.4: Contours of K_Q in J-β space for the KCS propeller with different sets of training data, with black dots illustrating the different samples used to train the model. Linear regression response surface is shown on the left. Gaussian process regression with σ_n=0.0001 predicted response surface is shown on the right.

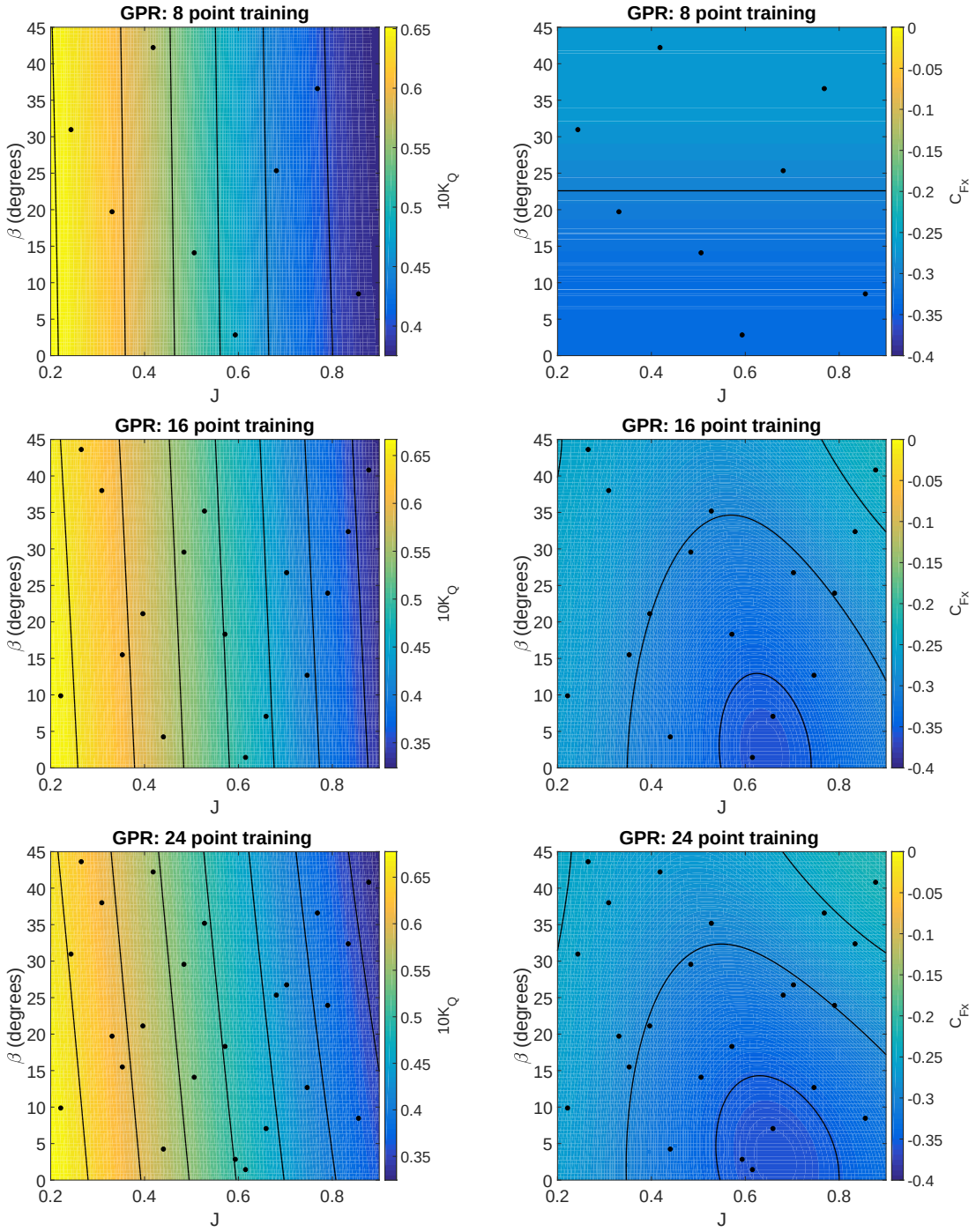

Figure A.5: Contours of K_Q and C_{Fx} in J-β space for the KCS propeller and rudder with different sets of training data, with black dots illustrating the different samples used to train the model. All results are for Gaussian process regression with a calculated σ_n. The left column of images show K_Q while the right column shows C_{Fx}. More training points are used for each subsequent row.

Table A.9: Linear regression coefficients for K_T, K_S, C_{Fx}, C_{Fy}, and C_{Mz} using the G2 eight point LHS training set with the Spalart Allmaras turbulence model.

	K_T	-10K_S	10K_Q	C_{Fx}	C_{Fy}
C_1	0.5345290305	0.3975592637	0.7934156768	0.1471394965	-0.0601792772
C_2	-0.4226926937	-1.0342701003	-0.5503789314	-1.3098336618	1.1571454275
C_3	0.0808059427	0.7560195931	0.1239172574	0.8802890611	-0.6526409694
C_4	0.0041557117	-1.1229502889	0.0032110496	-0.5136129952	0.5787844774
C_5	0.0199300119	0.9598115597	0.0323423376	0.3165818193	-0.4476160331
C_6	-0.1490251734	1.2731303479	-0.1657695166	0.6611638490	-0.8368780001

Table A.10: Linear regression coefficients for K_T, K_S, C_{Fx}, C_{Fy}, and C_{Mz} using the G2 eight point LHS training set with the $k-\omega$ SST turbulence model.

	K_T	-10K_S	10K_Q	C_{Fx}	C_{Fy}
C_1	0.4922036967	0.3742254919	0.6834371641	-0.2114225631	0.1431476845
C_2	-0.2702026946	-0.9635846231	-0.2079224697	-0.2320950820	0.4832225913
C_3	-0.0237223366	0.5468384169	-0.1087395397	0.1592390946	-0.2449764051
C_4	0.1671361242	-0.9812069765	0.2837225808	-0.0048276432	0.4118739945
C_5	-0.1806143171	0.5356170570	-0.2703055103	0.0991507996	-0.4979861139
C_6	-0.1813468948	1.8700159624	-0.2793355678	0.1452838384	-0.4515394650

A.4 Coefficients for Linear Regression Based Data-Driven Models

The coefficients for the data-driven model are presented here. For the feature vector [1 J J^2 β β^2 $J\beta$], the corresponding coefficients are denoted [C_1 C_2 C_3 C_4 C_5 C_6].

Table A.11: Linear regression coefficients for K_T, K_S, C_{Fx}, C_{Fy}, and C_{Mz} using the G3 eight point LHS training set with the Spalart Allmaras turbulence model.

C_1	0.552752337	0.36327798	0.7968946	0.258101568	-0.259840451
C_2	-0.418277763	-0.901201585	-0.504391235	-1.573165635	1.681390292
C_3	0.052683471	0.550472412	0.064430711	1.000354382	-0.830780534
C_4	-0.079393167	-1.086120399	-0.062284774	-0.68709302	0.948868651
C_5	0.098568493	0.896192311	0.10790787	0.404199133	-0.580375436
C_6	-0.139083741	1.340375315	-0.202987519	0.838867534	-1.400438693

APPENDIX B

Supplementary Tables and Figures for ONR Tumblehome Analysis

In this appendix are supplementary tables and figures for the ONR Tumblehome analysis.

B.1 ONR Tumblehome Latin Hypercube Sample points.

Table B.1: Values for each of the LHS points for ONR Tumblehome analysis for all LHS point.

LHS point	J	$\|\beta\|$ (degrees)	u (m/s)	v (m/s)
1	0.850	30.625	0.813	-0.481
2	0.650	21.875	0.622	-0.250
3	1.050	13.125	1.004	-0.234
4	1.250	4.375	1.195	-0.091
5	1.100	19.687	1.052	-0.376
6	1.000	2.188	0.956	-0.037
7	0.600	24.063	0.574	-0.256
8	0.800	32.812	0.765	-0.493
9	1.200	28.438	1.147	-0.621
10	0.900	10.937	0.861	-0.166
11	0.700	15.313	0.669	-0.183
12	1.300	6.562	1.243	-0.143
13	0.825	3.281	0.789	-0.045
14	0.675	29.531	0.645	-0.366
15	1.275	25.156	1.219	-0.573
16	1.125	20.781	1.076	-0.408
17	1.175	9.844	1.124	-0.195
18	0.975	14.219	0.932	-0.236
19	0.775	16.406	0.741	-0.218
20	0.575	7.656	0.550	-0.074
21	0.625	33.906	0.598	-0.402
22	0.925	27.344	0.884	-0.457
23	0.875	22.969	0.837	-0.355
24	1.075	5.469	1.028	-0.098
25	1.025	1.094	0.980	-0.019
26	1.325	31.719	1.267	-0.783
27	0.725	12.031	0.693	-0.148
28	1.225	18.594	1.171	-0.394

B.2 Supplementary Tables for Discretization Error Discussion

Table B.2 and Table B.3 show the values for each LHS point, the prediction of K_S for each of the grids as well as the percent error with respect to the G3 value for the G2 grid and the G1 grid. On average the absolute value of the percent error between the G3 and G2 grid for the K_S on the port side is 8.52% and 6.65% on the starboard side. The maximum absolute value of the percent difference between the G3 and G2 grids is over 28%, which occurs when the value of K_S is smallest on the starboard side, which correlates to LHS point six. At the same LHS point, the error between the G3 and G1 grids for the starboard side prediction of K_S is over 60%. LHS point six correlates to a relatively high J and low β. This case only affects the data-driven model briefly in the maneuver when the vessel first begins the maneuver as the rudder is deflected and the vessel begins to turn. On average the absolute value of the percent error between the G3 and G1 grid for K_S is larger than the error between the G3 and G2 grids, with 18.0% error on the port side and 11.5% on the starboard side. Similar to the prediction of K_X, the error tends to be larger on the port side than the starboard side.

Table B.4 and Table B.5 show the values of C_{Fx} for each LHS point and each grid as well as the percent error with respect to the G3 value for the G2 grid and the G1 grid. The $\overline{|\epsilon_{23}|}$ for the C_{Fx} on the port side is 4.38% and 7.50% on the starboard side. The $\overline{|\epsilon_{13}|}$ for the C_{Fx} on the port side is 7.79% and 7.03% on the starboard side. Overall there less error between the finest grid G3 and medium grid G2, however, the starboard side G1 prediction of C_{Fx} is in better agreement with the G3 grid prediction compared to the G2 grid prediction on average. The maximum percent error between the G3 and G2 grid is 12.04% on the port side and 15.62% on the starboard side. The maximum percent error between the G3 and G1 grid is larger with 16.77% error

Table B.2: Discretization error between different grids for ONR Tumblehome port side K_S.

LHS point	G1 (K_S)	G2 (K_S)	G3 (K_S)	ϵ_{23} (%)	ϵ_{13} (%)
1	0.053	0.057	0.059	2.179	9.667
2	0.037	0.041	0.037	-9.068	0.415
3	0.041	0.042	0.042	-1.904	1.741
4	0.034	0.036	0.032	-11.954	-5.209
5	0.068	0.071	0.064	-11.508	-6.011
6	0.006	0.006	0.005	-11.583	-5.788
7	0.039	0.042	0.036	-15.303	-7.480
8	0.044	0.051	0.051	0.533	14.289
9	0.078	0.077	0.072	-6.402	-7.326
10	0.028	0.030	0.034	12.689	18.024
11	0.020	0.023	0.021	-10.698	5.584
12	0.041	0.041	0.038	-8.412	-6.939

Table B.3: Discretization error between different grids for ONR Tumblehome starboard side K_S.

LHS point	G1 (K_S)	G2 (K_S)	G3 (K_S)	ϵ_{23} (%)	ϵ_{13} (%)
1	0.122	0.125	0.124	-0.378	2.089
2	0.066	0.068	0.068	0.073	3.743
3	0.054	0.055	0.056	1.732	4.389
4	0.010	0.009	0.013	25.506	20.684
5	0.088	0.090	0.089	-1.326	1.461
6	0.003	0.006	0.008	28.611	60.784
7	0.069	0.073	0.072	-0.945	3.426
8	0.126	0.129	0.130	0.573	2.741
9	0.150	0.151	0.149	-0.989	-0.312
10	0.040	0.041	0.044	5.526	8.630
11	0.051	0.052	0.054	3.681	6.833
12	0.021	0.025	0.028	10.429	23.329

Table B.4: Discretization error between different grids for ONR Tumblehome port side C_{Fx}.

LHS point	G1 (C_{Fx})	G2 (C_{Fx})	G3 (C_{Fx})	ϵ_{23} (%)	ϵ_{13} (%)
1	-0.239	-0.246	-0.246	0.160	2.765
2	-0.285	-0.280	-0.273	-2.479	-4.191
3	-0.262	-0.288	-0.283	-1.901	7.525
4	-0.296	-0.312	-0.342	8.606	13.409
5	-0.231	-0.237	-0.243	2.247	4.653
6	-0.290	-0.328	-0.348	5.838	16.773
7	-0.260	-0.246	-0.236	-4.064	-9.800
8	-0.248	-0.245	-0.245	-0.251	-1.303
9	-0.203	-0.210	-0.239	12.037	15.135
10	-0.295	-0.315	-0.310	-1.631	4.756
11	-0.284	-0.301	-0.285	-5.740	0.388
12	-0.292	-0.309	-0.334	7.569	12.742

on the port side and 18.91 on the starboard side.

Table B.5: Discretization error between different grids for ONR Tumblehome starboard side C_{Fx}.

LHS point	G1 (C_{Fx})	G2 (C_{Fx})	G3 (C_{Fx})	ϵ_{23} (%)	ϵ_{13} (%)
1	-0.149	-0.140	-0.144	2.630	-3.336
2	-0.193	-0.201	-0.208	3.505	7.179
3	-0.333	-0.308	-0.337	8.604	1.270
4	-0.306	-0.319	-0.377	15.617	18.908
5	-0.291	-0.258	-0.305	15.395	4.445
6	-0.322	-0.319	-0.371	14.053	13.239
7	-0.175	-0.180	-0.181	0.150	3.297
8	-0.122	-0.114	-0.116	1.612	-4.716
9	-0.204	-0.208	-0.222	6.262	8.146
10	-0.321	-0.297	-0.315	5.748	-1.937
11	-0.254	-0.231	-0.246	6.015	-3.573
12	-0.301	-0.315	-0.351	10.452	14.346

Table B.6: Discretization error between different grids for ONR Tumblehome port side C_{Mz}.

LHS point	G1 (C_{Mz})	G2 (C_{Mz})	G3 (C_{Mz})	ϵ_{23} (%)	ϵ_{13} (%)
1	0.349	0.372	0.384	3.140	9.202
2	0.459	0.459	0.448	-2.480	-2.483
3	0.346	0.382	0.377	-1.396	8.300
4	0.412	0.438	0.507	13.557	18.642
5	0.316	0.340	0.374	9.079	15.269
6	0.404	0.465	0.505	7.948	19.970
7	0.424	0.407	0.398	-2.262	-6.608
8	0.367	0.374	0.385	2.896	4.663
9	0.285	0.295	0.346	14.756	17.653
10	0.408	0.439	0.432	-1.702	5.520
11	0.447	0.460	0.442	-4.206	-1.230
12	0.402	0.423	0.465	9.019	13.517

Table B.7: Discretization error between different grids for ONR Tumblehome starboard side C_{Mz}.

LHS point	G1 (C_{Mz})	G2 (C_{Mz})	G3 (C_{Mz})	ϵ_{23} (%)	ϵ_{13} (%)
1	0.142	0.123	0.142	13.401	0.572
2	0.233	0.234	0.273	14.190	14.711
3	0.487	0.455	0.515	11.747	5.576
4	0.391	0.420	0.517	18.860	24.479
5	0.374	0.306	0.440	30.605	15.024
6	0.441	0.444	0.530	16.189	16.830
7	0.201	0.203	0.233	12.754	13.782
8	0.094	0.082	0.095	13.678	0.193
9	0.195	0.215	0.260	17.270	25.084
10	0.476	0.438	0.491	10.846	3.028
11	0.383	0.308	0.341	9.781	-12.204
12	0.388	0.429	0.473	9.336	17.815

B.3 Supplementary Tables for Turbulence Modeling Error Discussion

Table B.8 shows the difference between the prediction of K_S for the Spalart Allmaras turbulence model and the $k-\omega$ SST turbulence model for these LHS points for both the port and starboard side propellers. The average absolute value difference between the prediction of K_S between the two turbulence models is 16.23% on the starboard side and 18.76% on the port side. These numbers are somewhat skewed since the percent differences when small numbers are involved can be quite large. The K_S is significantly lower that the K_X and in the case of LHS point six, the $k-\omega$ SST turbulence model predicts a K_S of only 0.001, and the resulting percent error is nearly 80%. LHS points seven and eight are closest to the experimental steady turn parameters of J and β and provide more reasonable results on the starboard side. The K_S predicted by the $k-\omega$ SST turbulence model is 0.05% and 5.79% larger than the K_S predicted with the Spalart Allmaras turbulence model for LHS points seven and eight respectively. On the other hand the K_S predicted by the $k-\omega$ SST turbulence model

Table B.8: Turbulence modeling error between Spalart Allmaras and $k-\omega$ SST for K_S.

Side	LHS point	Spalart Allmaras (K_S)	$k-\omega$ SST (K_S)	ϵ_T (%)
Port	5	0.071	0.058	19.005
Port	6	0.006	0.005	15.845
Port	7	0.042	0.039	7.186
Port	8	0.051	0.067	-31.730
Port	9	0.077	0.100	-29.931
Port	10	0.030	0.025	14.794
Port	11	0.023	0.018	21.889
Port	12	0.041	0.037	9.670
Starboard	5	0.090	0.092	-1.560
Starboard	6	0.006	0.001	79.527
Starboard	7	0.073	0.073	-0.047
Starboard	8	0.129	0.136	-5.778
Starboard	9	0.151	0.160	-6.299
Starboard	10	0.041	0.037	9.477
Starboard	11	0.052	0.050	3.793
Starboard	12	0.025	0.019	23.401

is 7.19% smaller and 31.7% larger than the K_S predicted with the Spalart Allmaras turbulence model for LHS points seven and eight respectively. This illustrates the sensitivity of results to the turbulence model selection.

Table B.9 shows the differences between the two turbulence models prediction of C_{Fx} for each rudder. The average absolute value of the difference between the C_{Fx} predictions between the two turbulence models is 10.34% on the port side and 5.02% on the starboard side. The largest difference occurs at LHS point eight, for which the $k-\omega$ SST turbulence model prediction for the port side rudder has 27.64% less resistance than that predicted with the Spalart Allmaras turbulence model. This again illustrates how it is challenging to correctly model the propeller and rudder that operate in the wake of the hull.

Table B.10 shows the differences between the two turbulence models prediction of C_{Mz} for each rudder.

Table B.9: Turbulence modeling error between Spalart Allmaras and $k-\omega$ SST for C_{Fx}.

Side	LHS point	Spalart Allmaras (C_{Fx})	$k-\omega$ SST (C_{Fx})	ϵ_T (%)
Port	5	-0.237	-0.253	-6.664
Port	6	-0.328	-0.279	14.963
Port	7	-0.246	-0.238	3.164
Port	8	-0.245	-0.178	27.639
Port	9	-0.210	-0.227	-7.712
Port	10	-0.315	-0.279	11.522
Port	11	-0.301	-0.276	8.415
Port	12	-0.309	-0.317	-2.673
Starboard	5	-0.258	-0.269	-4.304
Starboard	6	-0.318	-0.316	0.666
Starboard	7	-0.180	-0.185	-2.743
Starboard	8	-0.114	-0.109	4.773
Starboard	9	-0.208	-0.194	6.545
Starboard	10	-0.296	-0.310	-4.444
Starboard	11	-0.231	-0.257	-11.477
Starboard	12	-0.315	-0.331	-5.176

Table B.10: Turbulence modeling error between Spalart Allmaras and $k-\omega$ SST for C_{Mz}.

Side	LHS point	Spalart Allmaras (C_{Mz})	$k-\omega$ SST (C_{Mz})	ϵ_T (%)
Port	5	0.340	0.372	-9.511
Port	6	0.465	0.402	13.560
Port	7	0.407	0.380	6.719
Port	8	0.374	0.291	22.208
Port	9	0.295	0.384	-30.278
Port	10	0.439	0.397	9.485
Port	11	0.460	0.409	11.117
Port	12	0.423	0.471	-11.355
Starboard	5	0.306	0.330	-7.845
Starboard	6	0.444	0.456	-2.654
Starboard	7	0.203	0.212	-4.312
Starboard	8	0.082	0.074	9.869
Starboard	9	0.215	0.195	9.233
Starboard	10	0.438	0.463	-5.900
Starboard	11	0.308	0.346	-12.423
Starboard	12	0.428	0.485	-13.088

B.4 Supplementary Figures for Gaussian Process Regression

This section includes supplementary figures showing the response surfaces predicted with Gaussian process regression.

B.4.1 σ_n of 0.0001 (near-zero)

Fig. B.1 shows the K_S for the propellers trained with the eight point sample set, the 16 point sample set, and the combination of the 16 point and 8 point sample set for a σ_n of 0.0001. The layout of the images corresponds to the same approach as Fig. 5.14. A smooth response surface is seen for all sets of training and are qualitatively similar to the response surfaces found with regression.

Fig. B.2 shows the C_{Fx} for the rudders (port side on the left and starboard side on the right) trained with the eight point sample set (in the top row), the 16 point sample set (in the middle row), and the combination of the 16 point and 8 point sample set (in the bottom row) for a σ_n of 0.0001. A response surface with lots of variability is seen for sets of training data with a large number of samples.

Fig. B.3 shows the C_{Fy} for the rudders for a σ_n of 0.0001 in a similar layout as the previous figure. A response surface with lots of variability is seen for sets of training data with a large number of samples.

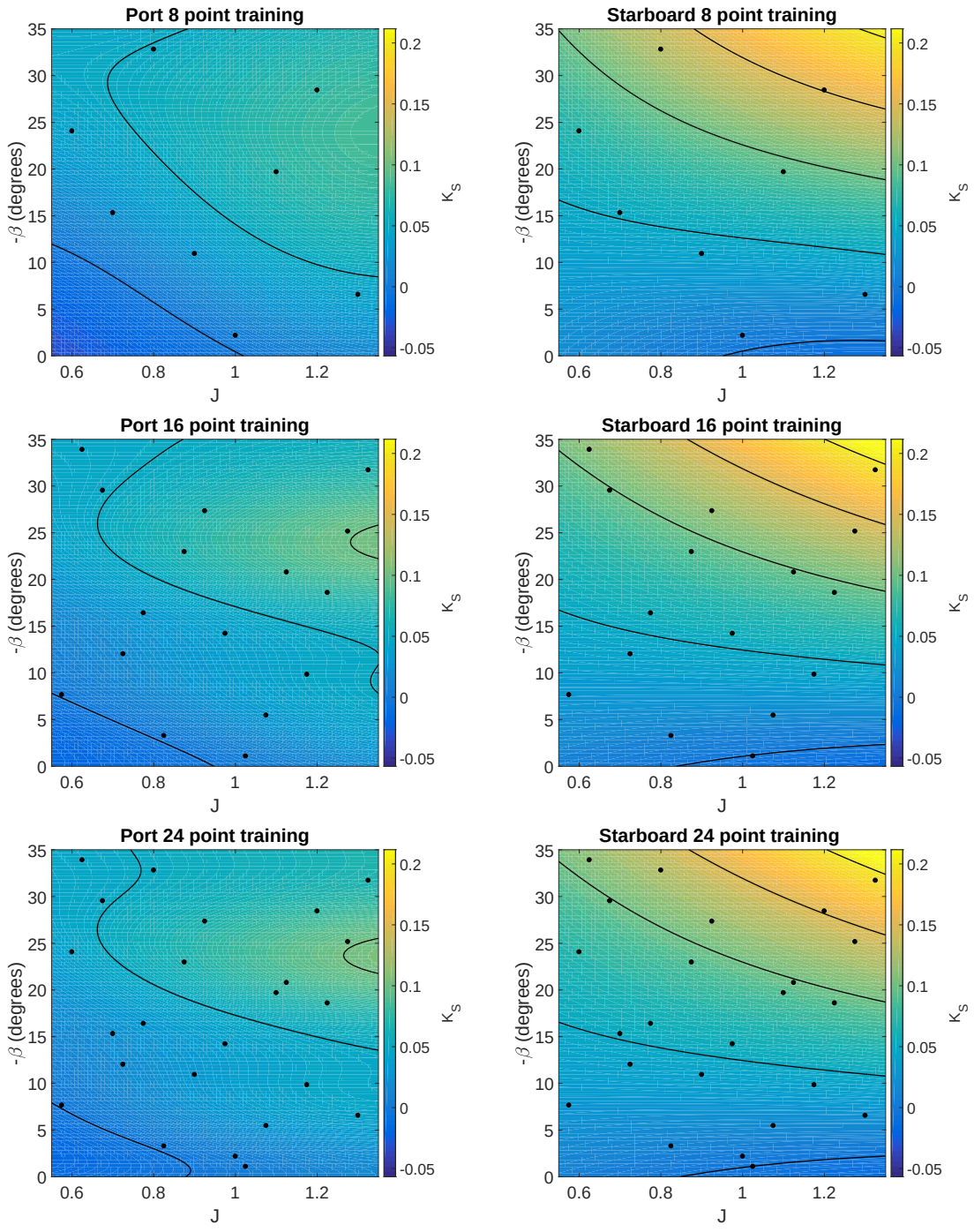

Figure B.1: Contours of K_S in J-β space predicted with Gaussian process regression with σ_n of 0.0001. The model using the 8-point LHS set is on the top. The model using the 16-point LHS set is in the middle row. The model using the combination of the 8-point and 16-point LHS set is on the bottom. The port side forces are shown on the left and the starboard forces are shown on the right.

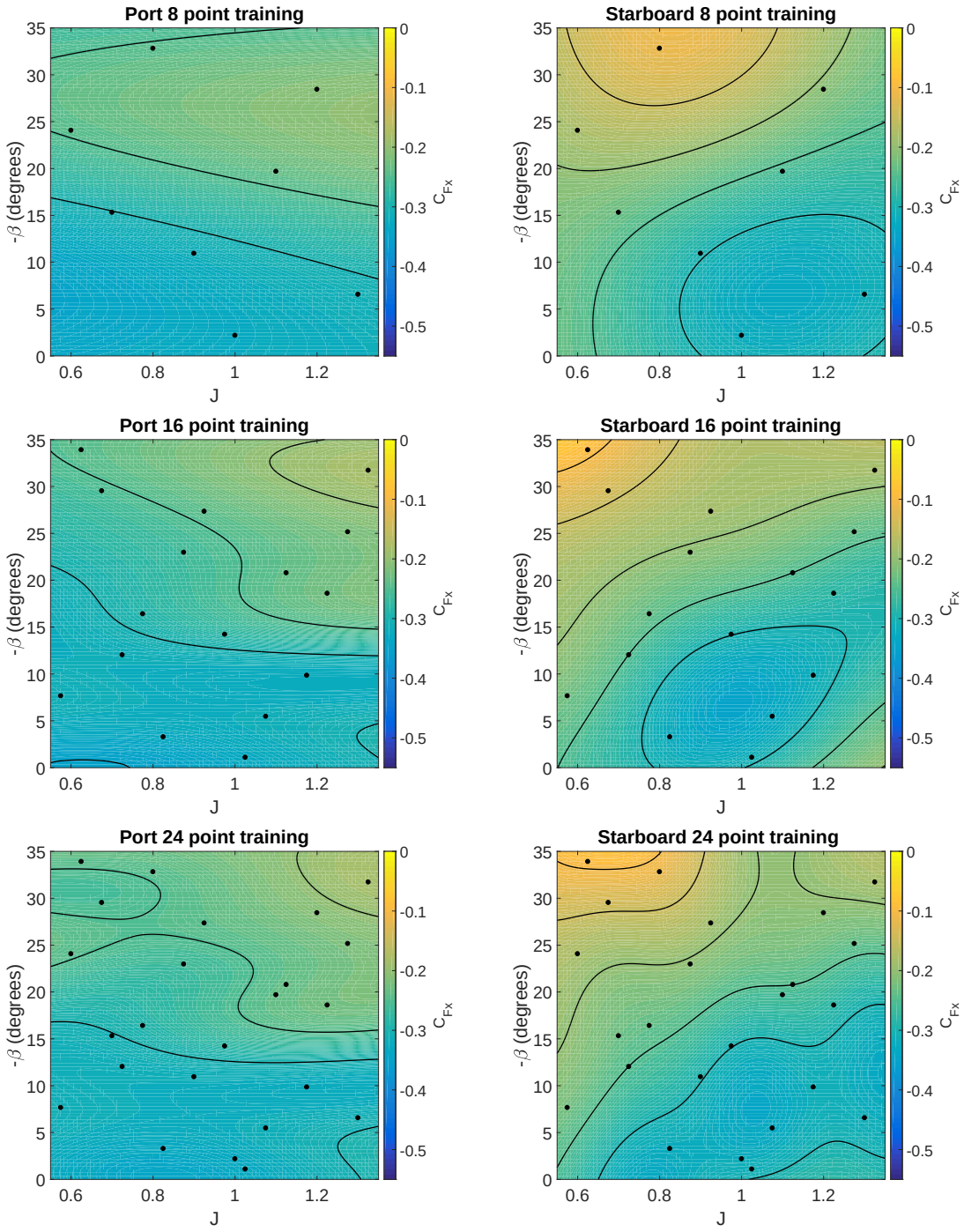

Figure B.2: Contours of C_{Fx} in J-β space predicted with Gaussian process regression with σ_n of 0.0001. The model using the 8-point LHS set is on the top. The model using the 16-point LHS set is in the middle row. The model using the combination of the 8-point and 16-point LHS set is on the bottom. The port side forces are shown on the left and the starboard forces are shown on the right.

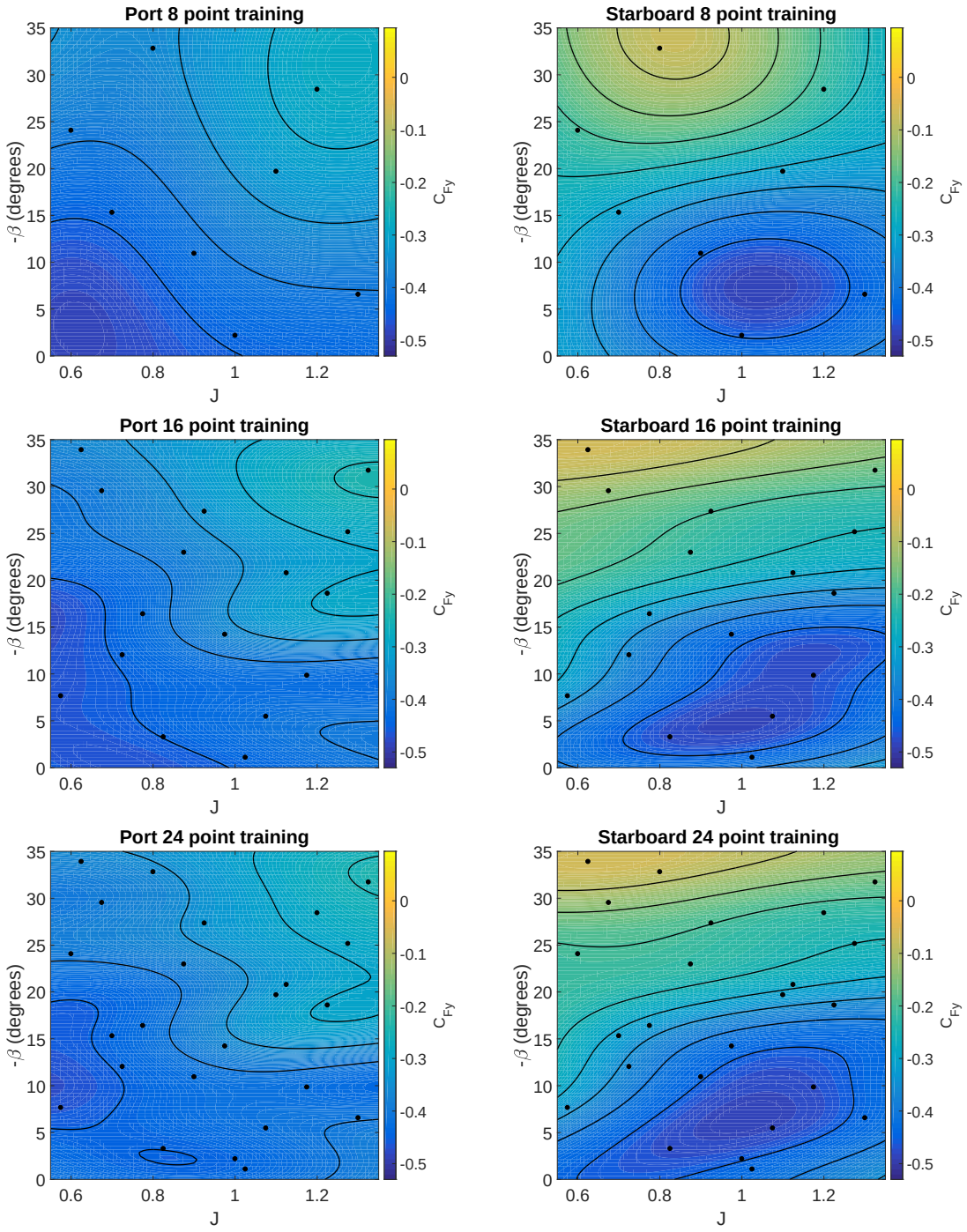

Figure B.3: Contours of C_{Fy} in J-β space predicted with Gaussian process regression with σ_n of 0.0001. The model using the 8-point LHS set is on the top. The model using the 16-point LHS set is in the middle row. The model using the combination of the 8-point and 16-point LHS set is on the bottom. The port side forces are shown on the left and the starboard forces are shown on the right.

B.4.2 σ_n calculated

Fig. B.4 shows the K_S for the propellers for a calculated σ_n of 0.022 for the port side propeller and for 0.010 for the starboard side propeller. The image layout for the figure is the same as the previous set of figures with the number of training sample points increasing with each row, and with the port and starboard side coefficients on the left and right respectively. A smooth response surface is seen for all sets of training.

Fig. B.5 shows the C_{Fx} for the propellers for a calculated σ_n of 0.057 for the port side rudder and for 0.023 for the starboard side rudder. The image layout for the figure is the same as the previous set of figures with the number of training sample points increasing with each row, and with the port and starboard side coefficients on the left and right respectively. A smooth response surface is seen for all sets of training.

Similar results are seen in Fig. B.6 which shows the C_{Fy} for the propellers for a calculated σ_n of 0.106 for the port side rudder and for 0.045 for the starboard side rudder. Again, a smooth response surface is seen for all sets of training.

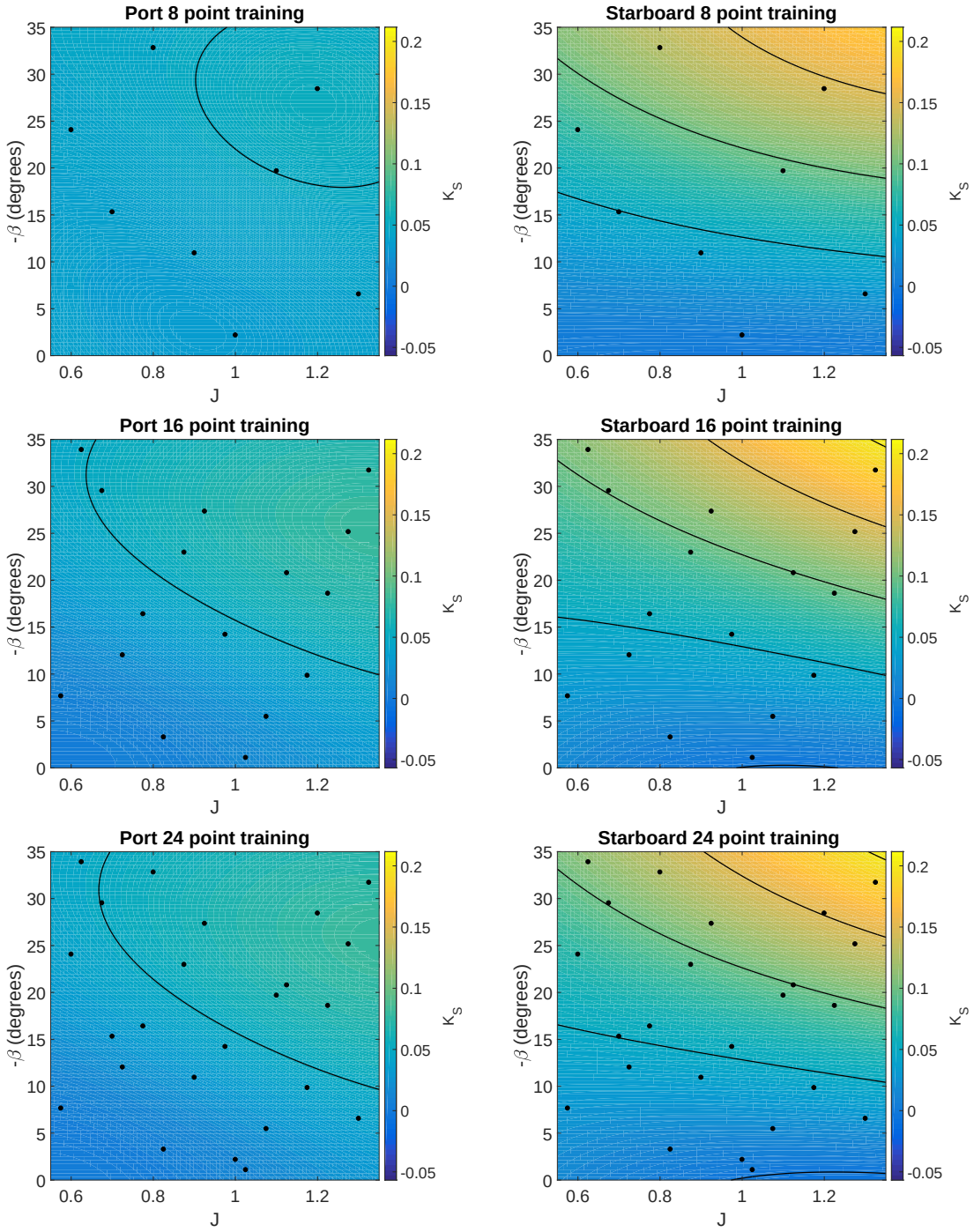

Figure B.4: Contours of K_S in J-β space predicted with Gaussian process regression with σ_n calculated. The model using the 8-point LHS set is on the top. The model using the 16-point LHS set is in the middle row. The model using the combination of the 8-point and 16-point LHS set is on the bottom. The port side forces are shown on the left and the starboard forces are shown on the right.

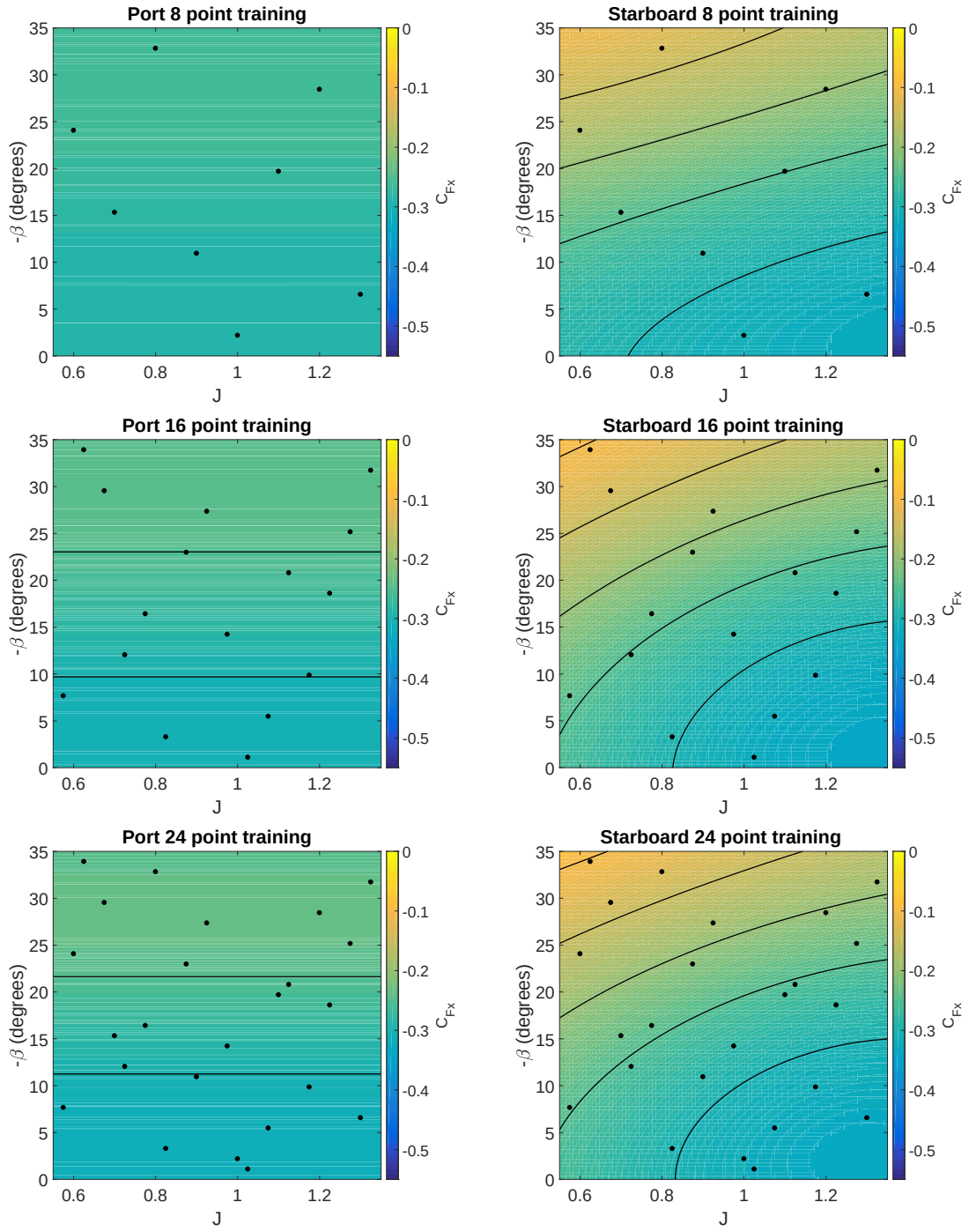

Figure B.5: Contours of C_{Fx} in J-β space predicted with Gaussian process regression with σ_n calculated. The model using the 8-point LHS set is on the top. The model using the 16-point LHS set is in the middle row. The model using the combination of the 8-point and 16-point LHS set is on the bottom. The port side forces are shown on the left and the starboard forces are shown on the right.

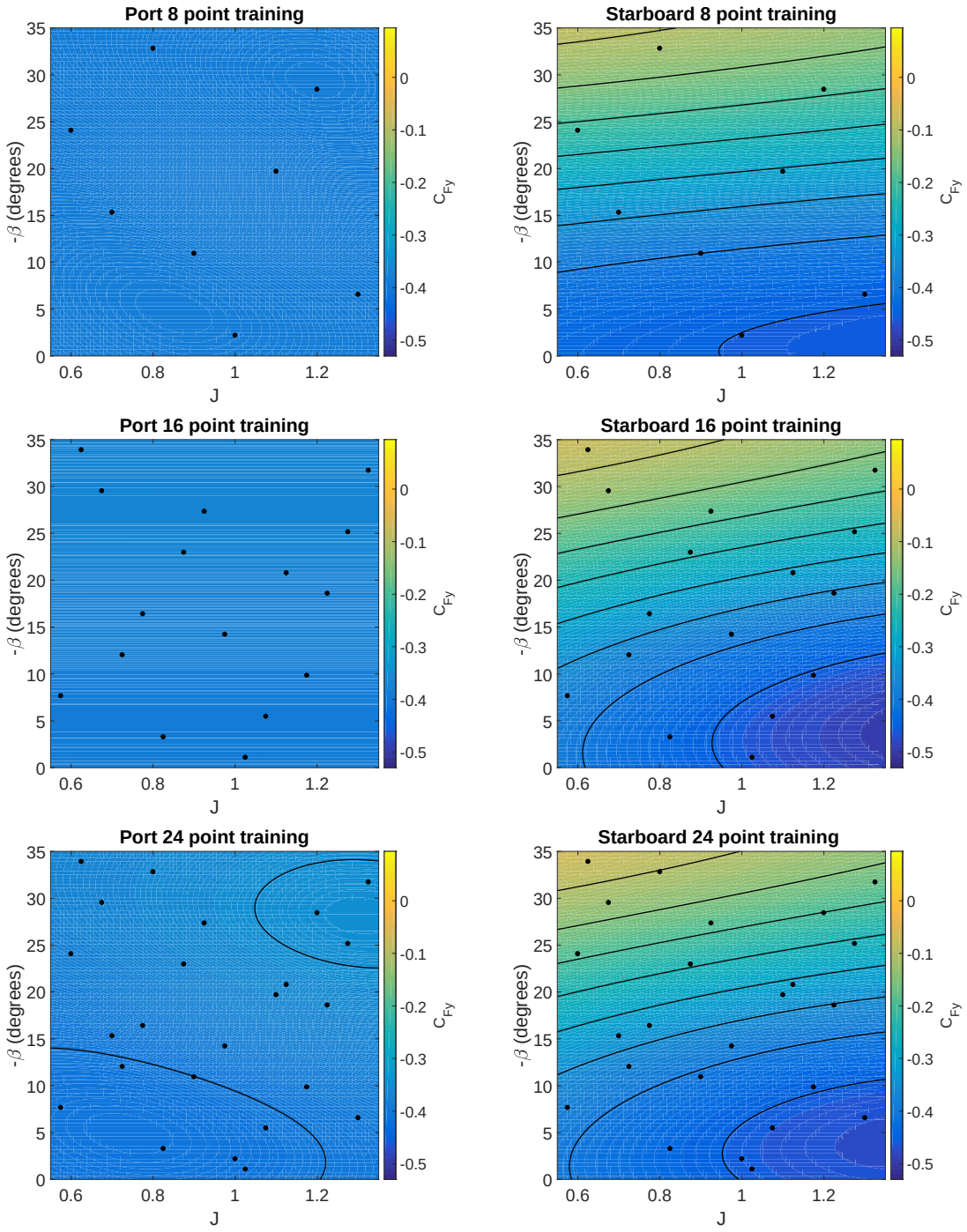

Figure B.6: Contours of C_{Fy} in J-β space predicted with Gaussian process regression with σ_n calculated. The model using the 8-point LHS set is on the top. The model using the 16-point LHS set is in the middle row. The model using the combination of the 8-point and 16-point LHS set is on the bottom. The port side forces are shown on the left and the starboard forces are shown on the right.

Table B.11: Linear regression coefficients for K_T, K_S, C_{Fx}, C_{Fy}, and C_{Mz} using the G2 eight point LHS training set with the Spalart Allmaras turbulence model.

	K_T	K_S	C_{Fx}	C_{Fy}	C_{Mz}
C_{p1}	0.738995787	-0.128710742	-0.247418435	-0.565757761	0.581403551
C_{p2}	-0.271740875	0.152146967	-0.224392458	0.100989001	-0.080774039
C_{p3}	-0.087155054	-0.032634583	0.119786783	-0.004505137	-0.007033881
C_{p4}	-0.561266578	0.38319183	0.134190991	0.191436376	-0.211143185
C_{p5}	0.284397848	-0.330712337	-0.107372029	-0.187535565	0.203974097
C_{p6}	0.194151747	-0.064648322	0.158670224	0.200522888	-0.208550824
C_{s1}	0.728931282	0.036215496	-0.202848515	-0.215941876	0.226893367
C_{s2}	-0.440469497	-0.044033337	-0.192556203	-0.478442768	0.446361167
C_{s3}	0.049498106	0.005386973	0.076970067	0.227659914	-0.215220485
C_{s4}	0.199946826	-0.057982587	0.175293678	0.261211913	-0.266461909
C_{s5}	-0.029877232	0.086577453	0.466039341	0.90778975	-0.879662291
C_{s6}	-0.109518742	0.281021988	-0.14042833	-0.170168985	0.172261932

B.5 Coefficients for Linear Regression Based Data-Driven Models

The coefficients for the data-driven model are presented here. For the feature vector $[1\ J\ J^2\ |\beta|\ |\beta|^2\ J|\beta|]$, the corresponding coefficients for the port side are denoted $[C_{p1}\ C_{p2}\ C_{p3}\ C_{p4}\ C_{p5}\ C_{p6}]$ and the corresponding coefficients for the starboard side are denoted $[C_{s1}\ C_{s2}\ C_{s3}\ C_{s4}\ C_{s5}\ C_{s6}]$.

Table B.12: Linear regression coefficients for K_T, K_S, C_{Fx}, C_{Fy}, and C_{Mz} using the G2 eight point LHS training set with the $k-\omega$ SST turbulence model.

	K_T	K_S	C_{Fx}	C_{Fy}	C_{Mz}
C_{p1}	0.852123115	0.036462121	-0.449481034	-0.989022406	1.021346472
C_{p2}	-0.496412987	-0.108896238	0.373795584	1.19441866	-1.221889623
C_{p3}	0.034030919	0.07248407	-0.206366354	-0.598191729	0.611675429
C_{p4}	-0.46622017	-0.006477109	-0.018696565	0.399552928	-0.40837637
C_{p5}	0.147132232	0.03759758	0.407475841	0.189241444	-0.208878338
C_{p6}	0.125101507	0.139286644	-0.032626451	-0.33012187	0.337805214
C_{s1}	0.854716796	0.047282684	-0.118662554	-0.050719228	0.062149785
C_{s2}	-0.590754206	-0.064769082	-0.314271863	-0.695318982	0.662854429
C_{s3}	0.092843152	0.010214522	0.119623997	0.270311947	-0.258456874
C_{s4}	-0.104409325	-0.111182744	-0.162978093	-0.376956175	0.361286541
C_{s5}	0.156966809	0.139963555	0.827796926	1.461026117	-1.424560186
C_{s6}	0.071370056	0.330950364	0.013295912	0.220477517	-0.21079638

Table B.13: Linear regression coefficients for K_T, K_S, C_{Fx}, C_{Fy}, and C_{Mz} using the G2 sixteen point LHS training set.

	K_T	K_S	C_{Fx}	C_{Fy}	C_{Mz}
C_{p1}	0.94548959	-0.10729674	-0.30082505	-0.56227055	0.58031468
C_{p2}	-0.77750445	0.12927434	-0.07800106	0.15338923	-0.14174365
C_{p3}	0.20855341	-0.02305283	0.03850260	-0.03723984	0.03091580
C_{p4}	-0.35605234	0.34468910	0.00755244	-0.02569846	0.01600840
C_{p5}	0.25271046	-0.28510930	0.01703792	0.07950932	-0.07019431
C_{p6}	-0.03391258	-0.06120494	0.20850353	0.24706461	-0.25993036
C_{s1}	0.82379001	0.04820346	-0.06417102	-0.21386999	0.21755721
C_{s2}	-0.57665595	-0.06565278	-0.41085190	-0.33407280	0.31759591
C_{s3}	0.09376016	0.01397503	0.14669462	0.06420191	-0.06079153
C_{s4}	0.06038774	-0.07665131	0.06242754	-0.00603974	0.00181378
C_{s5}	0.01227026	0.09223277	0.36537628	0.76679381	-0.74584272
C_{s6}	-0.00414796	0.29885395	0.02855733	0.22001389	-0.21314318

Table B.14: Linear regression coefficients for K_T, K_S, C_{Fx}, C_{Fy}, and C_{Mz} using the G3 eight point LHS training set.

	K_T	K_S	C_{Fx}	C_{Fy}	C_{Mz}
C_{p1}	0.75589936	-0.15569540	-0.36204429	-0.59852010	0.62171082
C_{p2}	-0.25046286	0.21731801	-0.02084615	0.15559644	-0.15155804
C_{p3}	-0.11073198	-0.06785736	0.00693630	-0.06230253	0.06049408
C_{p4}	-0.69072969	0.36731439	0.45429648	0.48500897	-0.51911256
C_{p5}	0.44248655	-0.29995330	-0.41433406	-0.52638533	0.55633012
C_{p6}	0.22464846	-0.07409698	0.02687844	0.09703435	-0.09872035
C_{s1}	0.74395868	0.03383845	-0.05517610	0.32592224	-0.30083130
C_{s2}	-0.46198040	-0.03383045	-0.52249864	-1.56175748	1.50163632
C_{s3}	0.05895599	0.00136333	0.21694023	0.72365341	-0.69856326
C_{s4}	0.15028068	-0.05869548	0.09459830	-0.36412710	0.34678918
C_{s5}	0.09488822	0.09619370	0.57045588	1.67079680	-1.63112584
C_{s6}	-0.13293063	0.26821930	-0.06571361	0.02595123	-0.01692304

BIBLIOGRAPHY

BIBLIOGRAPHY

Abramowski, T. (2005), Prediction of propeller forces during ship maneuvering, *Journal of Theoretical and Applied Mechanics*, *43*, 157–178.

Adwaith, N. (2020), Prediction of Ship Maneuverability, Master thesis.

Araki, M., H. Sadat-Hosseini, Y. Sanada, K. Tanimoto, N. Umeda, and F. Stern (2012), Estimating maneuvering coefficients using system identification methodswith experimental, system-based, and CFD free-running trial data, *Ocean Engineering*, *51*, 63–84.

Baltazar, J., and J. Rijpkema, D. Falcoa de Campos (2017), On the Use of the $\gamma - Re\theta$ Transition Model for the Prediction of the Propeller Performance at Model-Scale, in *Proceedings of Fifth International Symposium on Marine Propulsors*.

Bernitsas, M., D. Ray, and P. Kinley (1981), KT, KQ and Efficiency Curves for the Wageningen B-Series Propellers, Technical Report, The University of Michigan.

Bhattacharyya, A., V. Krasilnikov, and S. Steen (2016), A CFD-based scaling approach for ducted propellers, *Ocean Engineering*, *123*, 116–130.

Bouhlel, M., and J. Martins (2019), Gradient-enhanced kriging for high-dimensional problems, *Engineering with Computers*, *35*, 157–173.

Breuer, M., N. Jovii, and K. Mazaev (2003), Comparison of DES, RANS and LES for the separated flow around a flat plate at high incidence, *International Journal for Numerical Methods in Fluids*, *41*(4), 357–388.

Broglia, R., G. Dubbioso, D. Durante, and A. Di Mascio (2013), Simulation of turning circle by CFD: Analysis of different propeller models and their effect on manoevring prediction, *Applied Ocean Research*, *39*, 1–10.

Carrica, P. M., R. V. Wilson, R. W. Noack, and F. Stern (2007), Ship motions using single-phase level set with dynamic overset grids, *Computers & Fluids*, *36*(9), 1415–1433.

Carrica, P. M., H. Sadat-Hosseini, and F. Stern (2012), CFD analysis of broaching for a model surface combatant with explicit simulation of moving rudders and rotating propellers, *Computers & Fluids*, *53*, 117 – 132.

Celik, I., U. Ghia, P. Roache, C. Freitas, H. Coloman, and P. Raad (2008), Procedure for Estimation and Reporting of Uncertainty Due to Discretization in CFD Applications, *Journal of Fluids Engineering, ASME*, *130*(7).

Chu, Y., C. Lin, C. Hsin, and Y. Lee (2018), Numerical Analysis of Propulsion for Submarine with Highly Skewed Propeller, in *Proceedings of 7th PAAMES and AMEC*.

Cura-Hochbaum, A. (2006), Virtual PMM Tests for Manoeuvring Prediction, in *Proceedings of 26th Symposium on Naval Hydrodynamics*.

Dubbioso, G., R. Muscari, F. Ortolani, and A. Di Mascio (2017), Analysis of propeller bearing loads by CFD. Part I: Straight ahead and steady turning maneuvers, *130*, 241–259.

Eca, L., and M. Hoekstra (2009), Evaluation of numerical error estimation based on grid refinement studies with the method of the manufactured solutions, *Computers & Fluids*, *38*, 1580–1589.

Eca, L., G. Saraiva, G. Vaz, and H. Abreu (2015), The Pros and Cons of Wall Functions, in *Proceedings of the ASME 2015 34th International Conference on Ocean, Offshore, and Arctic Engineering*.

Eca, L., F. Pereira, and G. Vaz (2018), Viscous flow simulations at high Reynolds numbers without wall functions: Is $y^+ \simeq 1$ enough for the near-wall cells?, *Computers & Fluids*, *170*, 157–175.

Elshiekh, H. (2014), Maneuvering characteristics in calm water and regular waves for ONR Tumblehome , Master thesis.

Filip, G., D. Kim, S. Sahu, J. Kat, and K. Maki (2014), Bulbous bow retrofit of a container ship using an open-source Computational Fluid Dynamics (CFD) toolbox, *Transactions of the Society of Naval Architects and Marine Engineers*, *122*, 244–262.

Froude, R. (1889), On the part played in propulsion by differences of fluid pressure, *Transactions of the Institution of Naval Architects*, *30*, 390–405.

Froude, W. (1878), On the Elementary Relation Between Pitch, Slip,and Propulsive Efficiency, *Transactions of the Institution of Naval Architects*, *19*, 47–57.

Furst, J., J. Prihoda, and P. Straka (2013), Numerical simulation of transitional flows, *Computing*, *95*, S163–S182.

Gaggero, S., A. Coppede, D. Villa, G. Vernengo, and L. Bonfiglio (2019a), A Data-Driven Probabilistic Learning Approach for the Prediction of Controllable Pitch Propellers Performance, in *Proceedings of VIII International Conference on Computational Methods in Marine Engineering (Marine 2019)*.

Gaggero, S., G. Dubbioso, D. Villa, R. Muscari, and M. Viviani (2019b), Propeller modeling approaches for offdesign operative conditions, *Ocean Engineering*, *178*, 283–305.

Guo, H., and Z. Zou (2017), System-based investigation on 4-DOF ship maneuvering with hydrodynamic derivatives determined by RANS simulation of captive model tests, *Applied Ocean Research*, *68*, 11–25.

Guo, H., Z. Zou, Y. Liu, and F. Wang (2018), Investigation on hull-propeller-rudder interaction by RANS simulation of captive model tests for a twin-screw ship, *Ocean Engineering*, *162*, 259–273.

He, L., and S. Kinnas (2017), Numerical simulation of unsteady propeller/rudder interaction, *International Journal of Naval Architecture and Ocean Engineering*, *9*(6), 677–692.

Hirt, C., and B. Nichols (1981), Volume of fluid (VOF) method for the dynamics of free boundaries, *Journal of Computational Physics*, *1981*, 201–225.

Hoekstra, M. (2006), A RANS-based analysis tool for ducted propeller systems in open water condition, *International Shipbuilding Progress*, *53*, 205–227.

Hornik, K., M. Stinchcombe, and H. White (1989), Multilayer Feedfowrward Networks are Universal Approximators, *Neural Networks*, *2*, 359–366.

Hough, G., and D. Ordway (1965), The generalised actuator disc, *Developments in Theoretical and Applied Mechanics*, *2*, 317–336.

ITTC (2017), 1978 ITTC Performance Prediction Method, *International Towing Tank Conference (ITTC)-Recommended Procedures and Guidelines*, *7.5-02, 03-01.4*.

Jacobsen, N. G., D. R. Fuhrman, and J. Fredsoe (2011), A wave generation toolbox for the open-source cfd library: Openfoam®, *International Journal for Numerical Methods in Fluids*, *70*(9), 1073–1088.

Jin, Y., J. Duffy, S. Chai, and A. Magee (2019), DTMB 5415M dynamic manoeuvres with URANS computation using body-force and discretised propeller models, *Ocean Engineering*, *2019*, 305–317.

Kinnas, S., Y. Tian, and A. Sharma (2012), Numerical modeling of a marine propeller undergoing surge and heave motion, *International Journal of Rotating Machinery*, *2012*.

Knight, B., and K. Maki (2019), A semi-empirical multi-degree of freedom body force propeller model, *Ocean Engineering*, *178*, 270–282.

Knight, B., and K. Maki (2020), Multi-degree of freedom propeller force models based on a neural network and regression, *Journal of Marine Science and Engineering*, *8*(2), 89.

Knight, B., W. Xu, and K. Maki (2020), Numerical Prediction of Self-Propulsion in Extreme Head Seas, in *Proceedings of 33rd Symposium on Naval Hydrodynamics*.

Kose, K., A. Yumuro, and Y. Yoshimura (1981), Realization of maneuvering mathematical model, in *Proceedings of 3rd Symposium on Ship Maneuverability*, pp. 27–80, Society of Naval Architects of Japan (in Japanese).

Krige, D. (1951), A statistical approach to some mine valuation and allied problems on the Witwatersrand, Master thesis.

Kumar, P., and K. Mahesh (2016), Towards large eddy simulation of hull-attached propeller in crashback, in *Proceedings of the 31st Symposium on Naval Hydrodynamics*.

Langtry, R., and F. Menter (2009), Correlation-Based Transition Modeling for Unstructured Parallelized Computational Fluid Dynamics Codes, *AIAA Journal*, *47*(12), 2894–2906.

Laurenceau, J., and P. Sagaut (2008), Building Efficient Response Surfaces of Aerodynamic Functions with Kriging and Cokriging, *AIAA Journal*, *46*(2), 498–507.

Li, J., Y. Qu, Y. Chen, and H. Hua (2018), Investigation of added mass and damping coefficients of underwater rotating propeller using a frequency-domain panel method, *Journal of Sound and Vibration*, *432*, 602–620.

Liao, F., X. Yang, S. Wang, and G. He (2020), Grid-dependence study for simulating propeller crashback using large-eddy simulation with immersed boundary method, *Ocean Engineering*, *218*, 108,211.

Liem, R., C. Mader, and J. Martins (2015), Surrogate models and mixtures of experts in aerodynamic performance prediction for aircraft mission analysis, *Aerospace Science and Technology*, *43*, 126–151.

Mao, Y., and Y. L. Young (2016), Influence of skew on the added mass and damping characteristics of marine propellers, *Ocean Engineering*, *121*, 437–452.

Martio, J., A. Snchez-Caja, and T. Siikonen (2017), Open and ducted propeller virtual mass and damping coefficients by URANS-method in straight and oblique flow, *Ocean Engineering*, *130*, 92–102.

Matheron, G. (1963), Principles of Geostatistics, *Economic Geology*, *58*, 1246–1266.

Menter, F., and T. Esch (2001), Elements of Industrial Heat Transfer Predictions, in *Proceedings of the 16th Brazilian Congress of Mechanical Engineers*, vol. 20, pp. 117–127, COBEM.

Menter, F., M. Kuntz, and R. Langtry (2003), Ten years of industrial experience with the SST turbulence model, in *Proceedings of the fourth international symposiumon turbulence, heat and mass transfer*, pp. 625–632, Begell House.

Mizzi, K., Y. K. Demirel, C. Banks, O. Turan, P. Kaklis, and M. Atlar (2017), Design optimisation of propeller boss cap fins for enhanced propeller performance, *Applied Ocean Research*, *62*, 210 – 222.

Mousaviraad, S. (2010), CFD prediction of ship response to extreme winds and/or waves, Ph.D. thesis, University of Iowa.

Ortolani, F., G. Dubbioso, R. Muscari, S. Mauro, and A. Di Mascio (2018), Experimental and Numerical Investigation of Propeller Loads in Off-Design Conditions, *Journal of Marine Science and Engineering*, *6*(45).

Parsons, M., and W. Vorus (1981), Added Mass and Damping Estimates for Vibrating Propellers, in *Proceedings of the Propellers Symposium Transactions SNAME*.

Parsons, M., W. Vorus, and E. Richard (1980), Added Mass and Damping of Vibrating Propellers, Technical Report, The University of Michigan.

Persson, T., R. Liefvendahl, Benso, and C. Fureby (2006), Numerical investigation of the flow over an axisymmetric hill using LES, DES, and RANS, *Journal of Turbulence*, *7*.

Phillips, A., S. Turnock, and M. Furlong (2010), Accurate Capture of Propeller-Rudder Interaction using a Coupled Blade Element Momentum-RANS Approach, *Ship Technology Research*, *57:2*, 128–139.

Piro, D., P. White, B. Knight, K. Maki, and D. Kring (2020), Prediction of Ship Maneuvering in Waves Using a Combined RANS and Potential Flow Approach, in *Proceedings of 33rd Symposium on Naval Hydrodynamics*.

Prechelt, L. (1998), Early Stopping - But When?, in *In: Orr G.B., Mller KR. (eds) Neural Networks: Tricks of the Trade. Lecture Notes in Computer Science, vol 1524. Springer, Berlin, Heidelberg.*

Rankine, W. (1865), On the mechanical principles of the action of propellers, *Transactions of the Institution of Naval Architects*, *6*, 13–39.

Rasmussen, C., and C. Williams (2006), Gaussian Process for Machine Learning, in *MIT Press*.

Ribner, H. (1943), Propellers in yaw, Technical Report, National Advisory Committee for Aeronautics (NACA).

Roddy, R., D. Hess, and W. Falleer (2008), Utilizing Neural Networks to Predict Forces and Moments on a Submarine Propeller, in *46th AIAA Aerospace Sciences Meeting and Exhibit*.

Sacks, J., W. Welch, T. Mitchell, and H. Wynn (1989), Design and analysis of computer experiments, *Statistical Science*, *4*(4), 409–435.

Sadat-Hosseini, H., P. Wu, and F. Stern (2014), CFD Simulations of KVLCC2 Maneuvering with Diferent Propeller Modeling, in *(Proceedings of SIMMAN2014 Workshop)*.

Sanada, Y., H. Elsheikh, and F. Stern (2019), ONR Tumblehome course keeping and maneuvering in calm water and waves, *Journal of Marine Science and Technology*, *24*, 948–967.

Shen, Z., D. Wan, and P. M. Carrica (2015), Dynamic overset grids in OpenFOAM with application to KCS self-propulsion and maneuvering, *Ocean Engineering*, *108*, 287 – 306.

SIMMAN (2008), Workshop on Verification and Validation of Ship Manoeuvring Simulations.

SIMMAN (2020), Workshop on Verification and Validation of Ship Manoeuvring Simulations, in *(Workshop delayed due to Covid-19)*.

Singh, A., S. Medida, and K. Duraisamy (2017), Machine-Learning-Augmented Predictive Modeling of Turbulent Separated Flows over Airfoils, *AIAA Journal*, *55*(7), 2215–2227.

Spalart, P., and S. Allmaras (1994), A One-Equation Turbulence Model for Aerodynamic Flows, *Recherche Aerospatiale*, (1), 5–21.

Tang, J., Y. Hu, and B. Song (2017), Unsteady Aerodynamic Optimization of Airfoil for Cycloidal Propellers Based on Surrogate Model, *Journal of Aircraft*, *54*(4), 1241–1256.

Tezdogan, T., Y. K. Demirel, P. Kellett, M. Khorasanchi, A. Incecik, and O. Turan (2015), Full-scale unsteady RANS CFD simulations of ship behaviour and performance in head seas due to slow steaming, *Ocean Engineering*, *97*, 186 – 206.

Trodden, D., M. Woodward, and M. Atlar (2016), Accounting for ship manoeuvring motion during propeller selection to reduce CO2 emissions, *Ocean Engineering*, *123*, 346–356.

Umeda, N., A. Matsuda, H. Hashimoto, S. Yamamura, and A. Maki (2008), Model Experiments on Extreme Motions of a Wave-Piercing Tumblehome Vessel in Following and Quartering Waves, *Journal of the Japan Society of Naval Architects and Ocean Engineers*, *8*, 123–129.

Verma, A., H. Jang, and K. Mahesh (2012), The effect of an upstream hull on a propeller in reverse rotation, *Journal of Fluid Mechanics*, *704*, 61–88.

Vesting, F., and R. Bensow (2014), On surrogate methods in propeller optimisation, *Ocean Engineering*, *88*, 214–227.

Wang, J., and D. Wan (2018), Direct Simulations of Turning Circle Maneuver in Waves Using RANS-Overset Method, in *Proceedings of the ASME 2018 37th International Conference on Ocean, Offshore and Arctic Engineering*.

Wang, J., W. Zhao, and D. Wan (2016), Free Maneuvering Simulation of ONR Tumblehome Using Overset Grid Method in naoe-FOAM-SJTU Solver, in *Proceedings of 31st Symposium on Naval Hydrodynamics*.

Wang, J., L. Zou, and D. Wan (2018), Numerical simulations of zigzag maneuver of free running ship in waves by RANS-Overset grid method, *Ocean Engineering*, *162*, 55 – 79.

Wang, X., and K. Walters (2012), Computational analysis of marine-propeller performance using transition-sensitive turbulence modeling, *Journal of Fluids Engineering*, *134* (071107-1).

Webster, J., W. Neu, and S. Brizzolara (2019), Reynolds Stress Transition Modeling for Marine Propellers at Low Reynolds Number, in *Proceedings of Sixth International Symposium on Marine Propulsors*.

White, P. (2020), A Hybrid Computational Framework for the Simulation of Ships Maneuvering In Waves, Ph.D. thesis, University of Michigan.

White, P., D. Piro, B. Knight, and K. Maki (2021), A hybrid numerical framework for simulation of ships maneuvering in waves, *Journal of Ship Research*, *(Accepted for publication)*.

White, P. F., B. G. Knight, G. P. Filip, and K. J. Maki (2019), Numerical Simulations of the Duisburg Test Case Hull Maneuvering In Waves, in *Proceedings of SNAME Maritime Convention*.

Winden, B. (2014), Powering performance of a self propelled ship in waves, Ph.D. thesis, University of Southhampton.

Xue, Y., Y. Liu, C. Ji, G. Xue, and S. Huang (2020), System identification of ship dynamic model based on gaussian process regression with input noise, *Ocean Engineering*, *216*, 107,862.

Yamazaki, R. (1968), On the propulsion theory of ships on still water introduction, *Memoirs of the Faculty of Engineering, Kyushu University*, *27*(4).

Yao, J. (2015), On the propeller effect when predicting hydrodynamic forces for manoevring using RANS simulations of captive model tests, Ph.D. thesis, Technical University of Berlin.

Zhang, Z., and K. Duraisamy (2015), Machine Learning Methods for Data-Driven Turbulence Modeling, in *22nd AIAA Computational Fluid Dynamics Conference*.