

4-2019

Global Estimation Methodology for Wave Adaptation Modular Vessel Dynamics Using a Genetic Algorithm

Marco Alan Schoener

Follow this and additional works at: <https://commons.erau.edu/edt>



Part of the [Mechanical Engineering Commons](#)

Scholarly Commons Citation

Schoener, Marco Alan, "Global Estimation Methodology for Wave Adaptation Modular Vessel Dynamics Using a Genetic Algorithm" (2019). *Dissertations and Theses*. 453.
<https://commons.erau.edu/edt/453>

This Thesis - Open Access is brought to you for free and open access by Scholarly Commons. It has been accepted for inclusion in Dissertations and Theses by an authorized administrator of Scholarly Commons. For more information, please contact commons@erau.edu.

GLOBAL ESTIMATION METHODOLOGY FOR WAVE ADAPTIVE MODULAR VESSEL
DYNAMICS USING A GENETIC ALGORITHM

by

Marco Alan Schoener

A Thesis Submitted to the College of Engineering Department of Mechanical Engineering in
Partial Fulfillment of the Requirements for the Degree of
Master of Science in Mechanical Engineering

Embry-Riddle Aeronautical University
Daytona Beach, Florida
April 2019

GLOBAL ESTIMATION METHODOLOGY FOR WAVE ADAPTIVE MODULAR VESSEL
DYNAMICS USING A GENETIC ALGORITHM

by

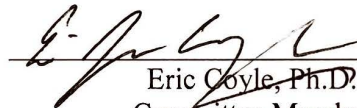
Marco Alan Schoener

This thesis was prepared under the direction of the candidate's Thesis Committee Chair, Dr. Patrick Currier, Associate Professor, Daytona Beach Campus, and Thesis Committee Members Dr. Eric Coyle, Associate Professor, Daytona Beach Campus, Dr. Charles Reinholtz, Professor, Daytona Beach Campus, and Dr. Yan Tang, Associate Professor, Daytona Beach Campus, and has been approved by the Thesis Committee. It was submitted to the Department of Mechanical Engineering in partial fulfillment of the requirements for the degree of Master of Science in Mechanical Engineering

Thesis Review Committee:



Patrick Currier, Ph.D.
Committee Chair



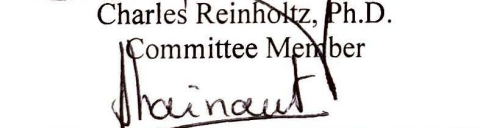
Eric Coyle, Ph.D.
Committee Member



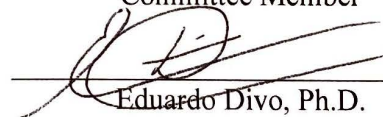
Charles Reinholtz, Ph.D.
Committee Member



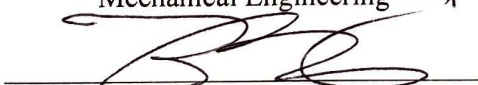
Yan Tang, Ph.D.
Committee Member



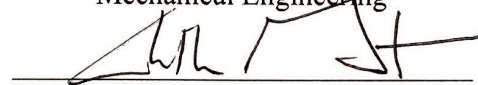
Jean-Michel Dhainaut, Ph.D.
Graduate Program Chair,
Mechanical Engineering



Eduardo Divo, Ph.D.
Department Chair,
Mechanical Engineering



Maj Mirmirani, Ph.D.
Dean, College of Engineering



Christopher Grant, Ph.D.
Associate Vice President of Academics

4/25/19
Date

Acknowledgments

This section is dedicated to the people who have helped me get to this point in my life. To help me achieve my goal of working/researching the field of robotics up to this level so far.

I want to thank, first and foremost, my mother (Isabel Helena Gouveia Schoener) and my sister (Hanalea Gouveia) for supporting me through the rough sections of college to be able to learn, grow, and seek opportunities that have allowed me to network and find work in the field that I love. I will never be able to thank them enough.

I would also like to thank my mentors that helped me grow through my academic and research career. Although it goes without saying that my committee were responsible for helping me at an academic level, I would like to point my attention here to thank Dr. Christopher Hockley, and soon-to-be Dr. Timothy Zuercher. These two have been there to help mentor and guide me to learn the implementation aspects of robotics that have kept the fire burning in me to continue. Dr. Hockley with his brilliant insight on how problems are approached theoretically with his experiences in the matter. And Timothy, whose jack-of-all-trades mastery in implementation of many different robotics aspects have immensely increased my own ability and confidence to continue robotics overall.

The rest go out to my best friends who have kept me sane and have pushed me to continue doing what I have a passion for. I would not feel comfortable growing without the initial support that these people have brought onto my life. I would like to thank the following people: Nicholas Ferraz, Luis Hill-Spranger, Damian Anderson, Natalie Oporto, Gabby Perez, Nicholas Middlebrooks, David J. “DJ” Thompson, and Stephen Cronin.

Abstract

Researcher: Marco Alan Schoener

Title: GLOBAL ESTIMATION METHODOLOGY FOR WAVE ADAPTIVE MODULAR VESSEL DYNAMICS USING A GENETIC ALGORITHM

Institution: Embry-Riddle Aeronautical University

Degree: Master of Science in Mechanical Engineering

Year: 2019

Determining parameters for a system model for marine vessels becomes more difficult as the model is made more complex. Work has been done to determine the equations of motion, but not to fully define how to estimate all of the system parameters. This work utilizes a global optimization methodology for estimating the system parameters using a genetic algorithm. The optimizer uses training data sets created from a set of ship maneuvering standards to minimize the error in the 3 degree-of-freedom equations of motion. The model has been optimized using a “No Surge-Yaw” model (minimal surge coupling) and a “Full” model (all states have coupling effects to each other) to determine how well each model can be estimated. The “No Surge-Yaw” model had the best results with making a working marine vessel model. The “Full” model was difficult to optimize due to the additional parameters that had unknown, nonlinear constraints. The “No Surge-Yaw” model was compared to linearized, no coupling version of the model that is commonly used. The linearized model vastly overestimated the results in sway and yaw rate motion while the “No Surge-Yaw” captured the expected coupling dynamics that do exist. Overall, the results of this methodology did generate a set of working marine vessel parameters for an unknown, coupled-state dynamic model.

Table of Contents

Abstract	iii
Table of Contents	iv
List of Tables	xi
List of Figures	xiii
Acronyms	xvi
Chapter I	1
1.1. Problem	1
1.2. Focus of Work	2
1.3. Thesis Contributions	2
Chapter II	3
2.1. System Model	3
2.1.1. Frames of Motion	3
2.1.2. Modeling Theories	5
2.1.3. Equations of Motion	6
2.1.4. Rigid-Body Motion	6
2.1.5. Hydrodynamic Motion	8
2.2. Disturbances	10
2.2.1. Current	10
2.2.2. Wind	11

2.3.	Thruster Model	11
2.3.1.	Throttle Mapping.....	12
2.3.2.	Open-Water Motor Model.....	13
2.4.	System Identification.....	14
2.4.1.	Modeling Techniques.....	14
2.4.2.	Optimization Techniques	15
2.4.3.	System Identification Boat Maneuvers	18
Chapter III	22
3.1.	Introduction	22
3.2.	Limitations and Assumptions	22
3.3.	Initial Parameter Estimation	23
3.3.1.	Rigid-Body	24
3.3.2.	Hydrodynamic.....	25
3.3.3.	Input Model Estimation.....	26
3.3.3.1.	Thruster Model Estimation.....	26
3.3.3.2.	Wind Model Estimation	27
3.3.4.	Maneuvering Testing.....	28
3.3.4.1.	Drift Test	28
3.3.4.2.	Steady-State Tests	29
3.3.4.3.	Stopping Test.....	29

3.3.4.4.	Turning Test	31
3.3.4.5.	Zig-zag Test.....	32
3.3.4.6.	Figure-Eight Test.....	33
3.4.	Genetic Algorithm Optimizer Setup.....	34
3.4.1.	Genetic Algorithm Data Sets.....	35
3.4.2.	Genetic Algorithm Settings.....	35
3.4.3.	Genetic Algorithm Error Function	35
3.4.4.	Genetic Algorithm Validation.....	36
3.5.	Equipment Used	37
3.5.1.	USV.....	37
3.5.2.	Propulsion.....	38
3.5.3.	Localization.....	39
3.6.	Testing Area	40
Chapter IV	42
4.1.	Overview	42
4.2.	Disturbances	42
4.3.	Initial Parameter Estimation Results	43
4.3.1.	Initial Surge Polynomial Fit	43
4.3.2.	Initial Yaw Rate Polynomial Fit.....	44
4.4.	Genetic Algorithm Fit (No Surge-Yaw Coupling).....	45

4.4.1.	GA “No Surge-Yaw” Fit #1	46
4.4.2.	GA “No Surge-Yaw” Fit #2	50
4.4.3.	GA “No Surge-Yaw” Fit #3	53
4.4.4.	GA “No Surge-Yaw” Validation Results.....	55
4.5.	Genetic Algorithm Fit (Full Coupling).....	58
4.5.1.	GA “Full” Fit.....	58
4.5.2.	GA “Full” Validation Results	61
4.6.	Comparison: “No Surge-Yaw” vs. “No Coupling”	63
4.7.	Down-Sampling Data Sets for Future Estimation	65
Chapter V		68
5.1.	Conclusion	68
5.2.	Future Work.....	69
Bibliography		71
Appendix.....		75
1.1.	Appendix 1 (To Be Added)	Error! Bookmark not defined.
Data Test Sets		75
2.1.	Wind Disturbance Data Sets.....	75
2.1.1.	Wind Set 1	75
2.1.2.	Wind Set 2.....	76
2.1.3.	Wind Set 3.....	77

2.2.	Spin Test Data Sets.....	78
2.2.1.	Spin Set 1	78
2.2.2.	Spin Set 2	80
2.3.	Stopping Test Data Sets.....	82
2.3.1.	Stopping Set 1	82
2.3.2.	Stopping Set 2	84
2.3.3.	Stopping Set 3	86
2.3.4.	Stopping Set 4	88
2.3.5.	Stopping Set 5	90
2.3.6.	Stopping Set 6	92
2.3.7.	Stopping Set 7	94
2.3.8.	Stopping Set 8	96
2.3.9.	Stopping Set 9	98
2.3.10.	Stopping Set 10.....	100
2.3.11.	Stopping Set 11.....	102
2.4.	Turning Test Data Sets	104
2.4.1.	Turning Set 1	104
2.4.2.	Turning Set 2.....	106
2.4.3.	Turning Set 3.....	108
2.4.4.	Turning Set 4.....	110

2.4.5.	Turning Set 5.....	112
2.4.6.	Turning Set 6.....	114
2.4.7.	Turning Set 7.....	116
2.5.	Zig-Zag Data Sets.....	118
2.5.1.	Zig-Zag Set 1.....	118
2.5.2.	Zig-Zag Set 2.....	120
2.5.3.	Zig-Zag Set 3.....	122
2.5.4.	Zig-Zag Set 4.....	124
2.5.5.	Zig-Zag Set 5.....	126
2.5.6.	Zig-Zag Set 6.....	128
2.5.7.	Zig-Zag Set 7.....	130
2.5.8.	Zig-Zag Set 8.....	132
2.5.9.	Zig-Zag Set 9.....	134
2.5.10.	Zig-Zag Set 10.....	136
2.5.11.	Zig-Zag Set 11.....	138
2.5.12.	Zig-Zag Set 12.....	140
2.5.13.	Zig-Zag Set 13.....	142
2.5.14.	Zig-Zag Set 14.....	144
2.5.15.	Zig-Zag Set 15.....	146
2.5.16.	Zig-Zag Set 16.....	148

2.5.17.	Zig-Zag Set 17	150
2.5.18.	Zig-Zag Set 18	152
2.6.	Figure Eight Data Sets	154
2.6.1.	Figure Eight Set 1	154
2.6.2.	Figure Eight Set 2	156
2.7.	Appendix 1 (To Be Added)	Error! Bookmark not defined.
	Genetic Algorithm Code	158
3.1.	GA Optimizer	158
3.2.	Genetic Algorithm Error Function	164
3.3.	Minion Boat Model	165

List of Tables

Table 1: Parameters set to be estimated and their units of the 2 nd -order Fossen model.	24
Table 2: The initial rigid-body parameters used in the estimation.	25
Table 3: The thruster ranges used to scale input forces/moment.	26
Table 4: List of the wind parameters needed to estimate wind diforces and moments.	28
Table 5: Throttle settings used for the surge and yaw drag tests.	29
Table 6: Throttle settings used for the stopping tests.	30
Table 7: Throttle settings used for the turning tests.	32
Table 8: Throttle settings used for the zig-zag tests.	33
Table 9: Throttle settings used for the figure-eight tests.	34
Table 10: the GA settings used with MATLAB 2018b.	35
Table 11: The weights used for the GA optimizer for each test’s data set.	36
Table 12: Minion rigid-body parameters.	38
Table 13: The surge-drag results used for initial estimation.	43
Table 14: The yaw-drag results used for initial estimation.	44
Table 15: Initial population to use for GA fitting.	46
Table 16: Parameters estimated from the Lake Ashby set of tests for Fit #1.	46
Table 17: Parameter limits used for the optimization for Fit #2.	49
Table 18: Parameters estimated from the Lake Ashby set of tests for Fit #2.	50
Table 19: Parameter limits used for the optimization for Fit #3.	52
Table 20: Parameters estimated from the Lake Ashby set of tests for Fit #3.	53
Table 21: The mean error of the four different validation tests using the past three parameter sets.	56
Table 22: Parameter set for the full coupling set model.	59

Table 23: The mean error of the four different validation tests between the “No Surge-Yaw” and “Full” coupling cases.	62
Table 24: Best set of parameters that fit the training and validation set of data.	63
Table 25: Mean error linearized coupling results.	65

List of Figures

Figure 1: Diagram of FRD frame on a marine vessel [2]. 3

Figure 2: Top view of the Minion USV16 with C.O. (yellow circle) and azimuth angle (δ) ranges for port and starboard displayed. 5

Figure 3: Example of using the Crossover Operator [19]. 17

Figure 4: Example of using the Mutation operator [19]. 17

Figure 5: A diagram of the standard stopping maneuver [22]. 19

Figure 6: A diagram of the standard stopping maneuver [22]. 20

Figure 7: A diagram of the standard zig-zag maneuver [22]. 21

Figure 8: Determine the relative velocity of the water due to global disturbance [21]. The current vector defines the {x,y,t} points to differentiate the average disturbance over time. 21

Figure 9: Top view of the Minion WAMV16 with C.O. (yellow circle) and azimuth angle (δ) ranges for port and starboard displayed. 25

Figure 10: Linear thrust curve for asymmetrical thrusters. 27

Figure 11: Input commands for a 50% stopping test. Both port and starboard are commanded simultaneously. 31

Figure 12: Input commands for a 70% turning test. The port thruster is commanded to stay on throughout the test. 32

Figure 13: Input commands for a 100% turning test. The thruster commands oppose each other every 10 deg change from the initial heading angle throughout the test. 33

Figure 14: A figure-eight test taken at Lake Ashby with noticeable drift due to wind. 34

Figure 15: Picture of the Minion USV. 37

Figure 16: Performance Data for the Copenhagen VM from Copenhagen’s data set at a hydrostatic state. The curves of interest are the “Thrust – Asymmetric Forward and Reverse” 39

Figure 17: Copenhagen Subsea VM Asymmetric thrusters.....	39
Figure 18: TORC Pinpoint GPS/INS (left) and WindSonic Anemometer (right).....	40
Figure 19: Google view of the test area, Lake Ashby.....	41
Figure 20: General wind disturbance at Lake Ashby. The overall USV speed was 0.5 m/s in 4 m/s winds.	42
Figure 21: Surge drag curve-fit from the steady-state test ranging from 0.75 to 2.5 m/s.....	43
Figure 22: Yaw drag curve-fit from the steady-state test ranging from -0.55 to 0.38 rad/s.	44
Figure 23: An example of a spin test yaw-rate states (Appendix 2.2.1).....	45
Figure 24: Surge result from a turning test validation set for Fit #1.....	47
Figure 25: Surge result from a stopping test validation set for Fit #1.	47
Figure 26: Sway and Yaw Rate result from a turning test validation set for Fit #1.	48
Figure 27: Sway and Yaw Rate result from a zig-zag test validation set for Fit #1.	49
Figure 28: Sway and Yaw Rate result from a turning test validation set for Fit #2.	51
Figure 29: Sway and Yaw Rate result from a zig-zag test validation set for Fit #2.	52
Figure 30: Sway and Yaw Rate result from a turning test validation set for Fit #3.	54
Figure 31: Sway and Yaw Rate result from a zig-zag test validation set for Fit #3.	55
Figure 32: A comparison of Fit #3 of the position validation figure-eight test. The green circle show initial position and the red circles show its ending positions for the real and simulated positions.	57
Figure 33: The figure-eight validation set using Fit #3's parameters.	58
Figure 34: Surge result from a figure-eight test validation set for the Full fit.....	59
Figure 35: Sway and Yaw Rate result from a zig-zag test validation set for the Full fit.....	60
Figure 36: Sway and Yaw Rate result from a turning test validation set for the Full fit.....	60

Figure 37: The full coupling simulation of the figure-eight validation test..... 62

Figure 38:Results of a zig-zag data set using the linearized EOMs and compared to the “No Surge-Yaw” #3 fit..... 64

Figure 39: Comparison of a zig-zag test of the 100Hz vs. the 10 Hz data set..... 66

Figure 40: Comparison of a zig-zag test of the 100Hz vs. the 1 Hz data set..... 66

Figure 41: Zoomed in differences in the 100Hz data set with 10 Hz and 1 Hz..... 67

Acronyms

X	Surge Force
Y	Sway Force
W	Heave Force
K	Roll Moment
M	Pitch Moment
N	Yaw Moment
N	North Position
E	East Position
Ψ	Yaw Angle
u	Surge Velocity
v	Sway Velocity
w	Heave Velocity
p	Roll Rate
q	Pitch Rate
r	Yaw Rate
m	Mass
C.G.	Center of Gravity
C.O.	Center of Origin
x	Forward Body-Axis
y	Right Body-Axis
z	Down Body-Axis
I_{xx}	Moment of Inertia about x-axis

I_{yy}	Moment of Inertia about y-axis
I_{zz}	Moment of Inertia about z-axis
I_{xy}	Moment of Inertia about xy-plane
I_{yz}	Moment of Inertia about yz-plane
I_{xz}	Moment of Inertia about xz-plane
N.D.	Non-Dimensional
GA	Genetic Algorithm
FRD	Front-Right-Down
NED	North-East-Down
DOF	Degree(s) of Freedom
USV	Unmanned Surface Vehicle
CAD	Computer Aided Design
WAMV	Wave Adaptive Modular Vessel
SNAME	Society of Naval Architects and Marine Engineers
AUVSI	Association for Unmanned Vehicle Systems International

Chapter I

Introduction

1.1. Problem

Modeling marine vessels has been a topic of interest since the 1960s with many different techniques and standards set in place to generate simulations and control laws. The works of Fossen were based on mostly large, monohulled vessels with rudder-based turning. With the increasing need for surveying and autonomy, the need for more adaptive types of marine vessels has increased. The vessel studied in this work is the WAMV-16, which is a platform that is designed to hold generally steady deck motion with independently, articulated pontoons. This vessel has been designed to handle sea-state conditions where other boats would have issues [1]. The WAMV-16 is the platform of choice for development and research in this thesis. It has been used exclusively in the AUVSI RobotX Maritime Challenge, where international collegiate teams compete using augmented WAMV USV16s to perform a broad array of boating and autonomy tasks.

A problem in modeling the WAMV is in part, that this vessel is more flexible than other most boats. No model exists for simulation and controls that captures its coupled-state dynamics. The coupled-state dynamics of a marine vessel is dependent on all the states, meaning that motion in one DOF is going to change the motion in another DOF. The coupling effects are not always intuitively determined with the standardized marine tests used in characterizing dynamic operation. A methodology needs to be created to estimate these parameters.

There have been works with [2] [3] [4] that made methodologies for generating system models for marine vessels. The latter two have done work on and for WAMV platforms. The models tend to be defined as rigid bodies with either linearized or highly coupled-state dynamics to accomplish desired motions and controlled states. The work with linearized dynamics [4] uses a WAMV-16 for model estimation and the highly coupled work [3] creates methodology for local

optimization estimations based on pre-existing, simulated Fossen models. The first work does not cover the coupled-state dynamics and the other does not describe how to get the initial parameters of a system model that needs to be further optimized.

1.2. Focus of Work

The focus of this thesis looks to approximate the coupled-state dynamics of a WAMV16 using a method to estimate the nonlinear dynamics. The ideal case is to have a model that can be used for simulation and controls work but needs a coupled-state estimation of the equations of motion to minimize expected error between simulated response and actual response of the WAMV.

1.3. Thesis Contributions

This work explored the following areas to present how the nonlinear dynamics of the WAMV can be determined and expressed:

- Developed a test methodology using a global optimizer for parameter identification to include capturing coupled-state dynamic terms.
- Generated a set of dimensional, coupled-state dynamic parameters for a WAMV16.

Chapter II

Literature Review

2.1. System Model

2.1.1. Frames of Motion

For modeling reference, there are two coordinates frames that are important: a body-fixed and global frame. The body-fixed frame is a right-handed Cartesian frame in \mathbb{R}^3 is aligned with the principal axes of inertia that results in the positive x-axis (forward/bow), the positive y-axis (right/starboard), and the positive z-axis (downward/towards the water). This is also known as the Front-Right-Down (FRD) frame. Velocities, forces, and moments are usually represented using this body-fixed frame (Figure 1). The forces and moments act on the vessel's center of gravity (C.G.), but a user-defined spot can be reference for the kinematic and dynamic states of the vessel for a constant, convenient reference known as the center of origin (C.O.).

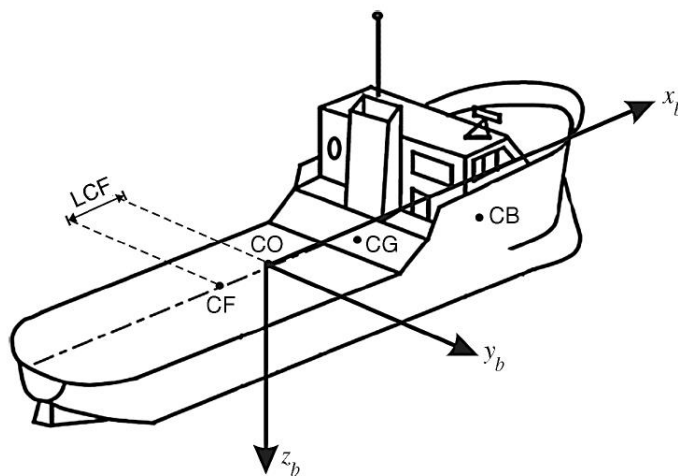


Figure 1: Diagram of FRD frame on a marine vessel [2].

The second frame is the inertial frame, which uses the NED frame representation. The NED frame is a transformation of the Earth Centered Earth Fixed (ECEF) frame, so the origin is on the surface of the earth and located nearby. NED is also a Cartesian frame in \mathbb{R}^3 with the x-axis positive toward north, the y-axis positive towards the east and the z-axis positive down. The NED

frame creates a 2-D plane on the curvature of the earth from a reference global position that is the origin of the frame. NED only approximates an inertial frame and the approximation gets worse the further from the origin you travel. The relatively small operating area of the vessel makes this approximation accurate. The NED frame is used to represent the positions and velocities of a system [2].

In a full six degree of freedom (6-DOF) model, a complete rotation matrix or quaternion would need to be used to transform between the vessel FRD and NED frames. However, because the model will be simplified to a three degree of freedom (3-DOF) planar representation, only a single rotation along the z-axis needs to be applied, Eqn. (3) and (4) [5]. The body-frame is denoted by the letter b and the inertial frame is denoted by the letter n . An unlabeled vector, \vec{v} , will be defaulted to \vec{v}^b .

$$\vec{v}^b = (u, v, r) \quad (1)$$

$$\vec{v}^n = (v_N, v_E, v_\psi) \quad (2)$$

$$R_b^n = \begin{bmatrix} \cos(\Psi) & -\sin(\Psi) & 0 \\ \sin(\Psi) & \cos(\Psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3)$$

$$\vec{v}^n = R_b^n \vec{v}^b \quad (4)$$

In this rotation, u and v are surge (forward) and sway (sideways) velocities respectively; v_N and v_E are velocity in the northing and easting directions respectively; and ψ is the yaw (heading) angle of the vehicle, measured off true north.

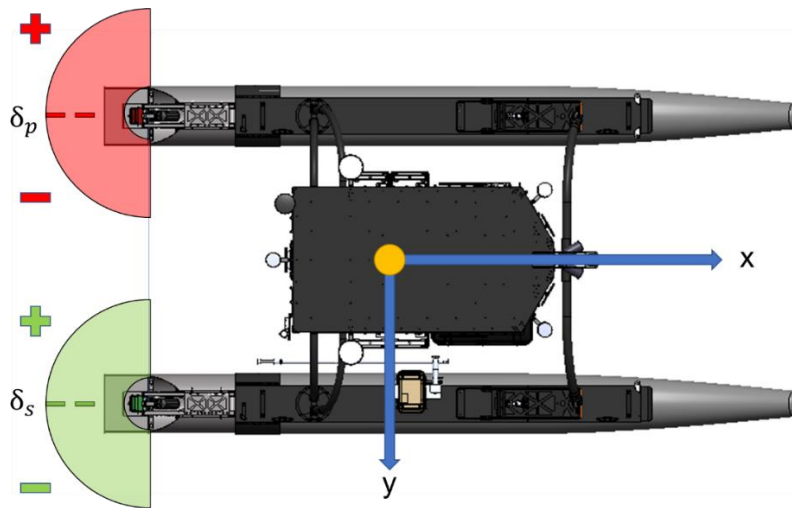


Figure 2: Top view of the Minion USV16 with C.O. (yellow circle) and azimuth angle (δ) ranges for port and starboard displayed.

2.1.2. Modeling Theories

In naval architecture, there are two classic model theories used to describe the motion of marine vessels as described by Fossen: Maneuvering Theory and Seakeeping Theory [2]. Maneuvering theory is used for vessels that operate in calm, restricting waters. The model is derived through moving at a positive forward speed with constant hydrodynamic parameters to define the added mass and damping on the vessel. Seakeeping theory studies the motion of vessels given wave excitation to determine operability in more sea-states than the maneuvering theory. This model is defined by maintaining its constant speed and heading and determines overall vessel state due to perturbations from the fixed equilibrium states, such as its speed and heading.

This thesis work will focus on the maneuvering vessel model due to its simplistic representation of the motion of the marine vessel. Other reference works use the same model to achieve the same goal of having an operable model simulation and controls work [3] [4] [6] [7].

2.1.3. Equations of Motion

The dynamic system model is setup between rigid-body kinetics and the hydrodynamics of each DOF with a control input to move the USV along. The general equation of motion (EOM) is defined as:

$$\mathbf{M}\dot{\mathbf{v}} + \mathbf{C}(\mathbf{v})\mathbf{v} + \mathbf{D}(\mathbf{v})\mathbf{v} = \boldsymbol{\tau} + \boldsymbol{\tau}_w \quad (5)$$

The system model in Eqn. (5) is depicted on the left side of the equation with the all the remaining control forces on the right side of the equation. The control forces encompass the forces and moments provided by the thrusters and the disturbances due to external effects such as wind and current.

The general equations of motion are given in [2]. Using this general form, the following subsections break the model into four components: rigid-body motion, hydrodynamic motion, disturbances, and actuator forces. Each section describes that aspect of the model's construction in detail and can be combined into a complete parameterized model of the vessel.

2.1.4. Rigid-Body Motion

The rigid-body representation defines the vessel's motion due to the inertia and kinetics. A 6-DOF model as given by [2] is shown in Eqn. (6)-(11).

$$X = m(\dot{u} - vr + wq - x_{CG}(q^2 + r^2) + y_{CG}(pq - \dot{r}) + z_{CG}(pr + \dot{q})) \quad (6)$$

$$Y = m(\dot{v} - wp + ur - y_{CG}(r^2 + p^2) + z_{CG}(qr - \dot{p}) + x_{CG}(qp + \dot{r})) \quad (7)$$

$$W = m(\dot{w} - uq + vp - z_{CG}(p^2 + q^2) + x_{CG}(rp - \dot{q}) + y_{CG}(rq + \dot{p})) \quad (8)$$

$$K = I_{xx}\dot{p} + (I_{zz} - I_{yy})qr + (\dot{r} + pq)I_{xz} + (r^2 - q^2)I_{yz} + (pr - \dot{q})I_{xy} \\ + m(y_{CG}(\dot{w} - uq + vp) - z_{CG}(\dot{v} - wp + ur)) \quad (9)$$

$$M = I_{yy}\dot{q} + (I_{xx} - I_{zz})rp - (p + qr)I_{xy} + (p^2 - r^2)I_{xz} + (qp - \dot{r})I_{yz} \\ + m(z_{CG}(\dot{u} - vr + wq) - x_{CG}(\dot{u} - uq + vp)) \quad (10)$$

$$\begin{aligned}
N = I_{zz}\dot{r} + (I_{yy} - I_{xx})pq - (\dot{q} + rp)I_{yz} - (q^2 - p^2)I_{xy} + (rq - \dot{p})I_{xz} \\
+ m(x_{CG}(\dot{v} - wp + ur) - y_{CG}(\dot{u} - vr + wq))
\end{aligned} \tag{11}$$

where:

m – Mass of the Vessel [kg]

I_{zz} – Mass of the Vessel [kgm^2]

X, Y, W – Surge, Sway, and Heave Forces [N]

K, M, N – Roll, Pitch, and Yaw Moments [Nm]

u, v, w – Surge, Sway, and Heave Speeds [m/s]

p, q, r – Roll, Pitch, Yaw Rates [rad/s]

As noted by Fossen [2], this can be represented in a vector form in Eqn. (12), where M is the mass matrix, C is the Coriolis matrix, τ is the actuator/disturbance forces and moments, and v is the state vector in the FRD frame.

$$M_{RB}\dot{v} + C_{RB}(v)v = \tau_{RB} \tag{12}$$

To simplify this initial set of equations, the following assumptions were applied:

- The roll, pitch, and heave states are neglected; this results in a horizontal-plane, 3-DOF model. Wave motion is not captured with this assumption.
- The x and y C.G. are reference at the vessel's denoted center of origin (C.O.)
- Coupling effects between the surge and yaw states exists.

Applying these assumptions leads to the following form of the model:

$$\vec{v} := \{u, v, r\}^T \tag{13}$$

$$\vec{\tau} = \{X, Y, N\}^T \tag{14}$$

$$M_{RB} = \begin{bmatrix} m & 0 & -my_{CG} \\ 0 & m & mx_{CG} \\ -my_{CG} & mx_{CG} & I_{zz} \end{bmatrix} \tag{15}$$

$$C_{RB} = \begin{bmatrix} 0 & 0 & -m(x_{CG}r + v) \\ 0 & 0 & -m(y_{CG}r - u) \\ m(x_{CG}r + v) & m(y_{CG}r - u) & 0 \end{bmatrix} \quad (16)$$

2.1.5. Hydrodynamic Motion

The hydrodynamic terms show the effects of moving the rigid-body through the water. All hydrodynamic coefficients used in the following models utilize the SNAME nomenclature that relates the force based on the related states denoted in its subscript. When dealing with nonlinear models, there are two methods for handling them: first principles and truncated-Taylor series expansions [2]. The first principles model uses well known, established physical properties of the vessel to make the model. This is significantly easier due to the number of parameters being smaller than the Taylor series. On the other hand, the truncated-Taylor series in Eqn. (17) defines a vessel that can be more easily determined experimentally through third-order odd functions to describe the vessel's damping [8]. The forces are related to all possible combinations of states to generate the Taylor-series function in the general form (Eqn. (19)). The terms can be difficult to experimentally fit, which is why a second-order modulus form [9] in Eqn. (18) is given that is motivated by the square law of damping for hydrodynamics and aerodynamics. The model will be defined by the second-order functions.

$$X = X_{\dot{u}}\dot{u} + X_u u + X_{uuu}u^3 + X_{uuv}u^2v + X_{urr}ur^2 + \dots \quad (17)$$

$$X = X_{\dot{u}}\dot{u} + X_u u + X_{u|u}|u| + X_{\dot{v}}\dot{v} + X_v v + X_{v|v}|v| + \dots \quad (18)$$

$$F(s) = \sum_{n_1=0}^{\infty} \dots \sum_{n_d=0}^{\infty} \frac{(s_1 - \bar{s}_1)^{n_1} \dots (s_d - \bar{s}_d)^{n_d}}{n_1! \dots n_d!} \left(\frac{\partial^{n_1+\dots+n_d} \vec{F}}{\partial s_1^{n_1} \dots \partial s_d^{n_d}} \right) (\bar{s}_1, \dots, \bar{s}_d) \quad (19)$$

where:

F – Forcing terms

d – Number of States (v)

s_i – State elements (from $0 \leq i \leq d$)

\bar{s}_i – Steady – state value around Taylor expansion

The terms that change this behavior are the added mass (AM) due to the Mass and Coriolis matrices and the hydrodynamic drag matrix, D . Eqn. (20) is the new 3-DOF vectorized equations of motion. The additional parameters are the added mass coefficients and the drag coefficients. The parameters are constant, and the equations are nonlinear.

$$M\dot{\vec{v}} + C(\vec{v})\vec{v} + D(\vec{v})\vec{v} = \vec{\tau} \quad (20)$$

The mass matrix (M), is an inertia tensor that includes both the rigid-body, M_{RB} , and added mass matrix, M_{AM} .

$$M = M_{RB} + M_{AM} \quad (21)$$

$$M_{AM} = \begin{bmatrix} X_{\dot{u}} & X_{\dot{v}} & X_{\dot{r}} \\ Y_{\dot{u}} & Y_{\dot{v}} & Y_{\dot{r}} \\ N_{\dot{u}} & N_{\dot{v}} & N_{\dot{r}} \end{bmatrix} \quad (22)$$

The Coriolis matrix is a tensor that describes the Coriolis effects in the EOM.

$$C(\vec{v}) = C_{RB} + C_{AM} \quad (23)$$

$$C_{AM}(\vec{v}) = \begin{bmatrix} 0 & 0 & Y_{\dot{u}}u + Y_{\dot{v}}v + Y_{\dot{r}}r \\ 0 & 0 & -X_{\dot{u}}u - X_{\dot{v}}v - X_{\dot{r}}r \\ -Y_{\dot{u}}u - Y_{\dot{v}}v - Y_{\dot{r}}r & X_{\dot{u}}u + X_{\dot{v}}v + X_{\dot{r}}r & 0 \end{bmatrix} \quad (24)$$

$$(25)$$

The drag matrix, $D(\vec{v})$ (Eqn. (26)), approximates the hydrodynamic drag the vessel experiences. The drag model is constructed of a set of linear terms, D_L , and nonlinear terms, D_{NL} . This results in a second order drag model cross-coupling surge to yaw and sway to yaw. The other off-terms, such as Y_{uu} or X_{rr} , are included to add more coupling effects to help correct weaker, but still important, coupling.

$$D(\vec{v}) = D_L + D_{NL}(\vec{v}) \quad (26)$$

$$D_L = \begin{bmatrix} X_u & X_v & X_r \\ Y_u & Y_v & Y_r \\ N_u & N_v & N_r \end{bmatrix} \quad (27)$$

$$D_{NL}(\vec{v}) = \begin{bmatrix} X_{uu}|u| & X_{vv}|v| & X_{rr}|r| \\ Y_{uu}|u| & Y_{vv}|v| + Y_{vr}|r| & Y_{rr}|r| + Y_{rv}|v| \\ N_{uu}|u| & N_{vv}|v| + N_{vr}|r| & N_{rr}|r| + N_{rv}|v| \end{bmatrix} \quad (28)$$

2.2. Disturbances

Disturbances affect the vessel's motion through water and ultimately its simulation for uses in controls and effects. There are two disturbances that are of potential concern: current and wind. The full 3-DOF equation of motion, including disturbances, is given by:

$$M\dot{\vec{v}} + C(\vec{v})\vec{v} + D(\vec{v})\vec{v} = \vec{\tau} + \vec{\tau}_d \quad (29)$$

2.2.1. Current

Current can create extra drag on the vessel that either hinders its motion (against the current) or increases it (with the current). The current \vec{v}_c can be considered both uniform and irrotational during operation (Eqn. (30)); meaning that the current effects on the body-fixed speeds change with respect to the angle of the current and the yaw angle of the vessel in the NED frame. If the rigid-body Coriolis and Mass matrices are parameterized independently of the linear velocity \vec{v} (vessel speed including current speed), then the rigid-body matrices equal its relative velocity \vec{v}_r in Eqn. (31) [2]. This means that subtracting the effects of the current speed from the vessel velocity shows the motion of the vessel as if there was no current (Eqn. (32)), leading to the effects of the hydrodynamic terms modeled as seen in Eqn. (33).

$$\vec{v}_c = \{u_c, v_c, 0\}^T \quad (30)$$

$$\vec{v}_r = \vec{v} - \vec{v}_{current} \quad (31)$$

$$M_{RB}\dot{\vec{v}} + C_{RB}(\vec{v})\vec{v} \equiv M_{RB}\dot{\vec{v}}_r + C_{RB}(\vec{v}_r)\vec{v}_r \quad (32)$$

$$(M_{RB} + M_{AM})\dot{\vec{v}}_r + (C_{AM}(\vec{v}_r) + C_{RB}(\vec{v}_r))\vec{v}_r + D(\vec{v}_r)\vec{v}_r = \vec{\tau} + \vec{\tau}_d \quad (33)$$

2.2.2. Wind

Wind disturbances affect vessels with surface areas above water by adding a drag term [2] [10]. This can be estimated by calculating the relative velocity of the wind with respect to the vehicle and using the resulting velocity to calculate aerodynamic drag. The terms for the wind forces are the relative wind velocity (V_{rw}), the apparent angle of attack (γ_{rw}), approximate air density (ρ_{air}), general surface area of front and side (A_{FW}, A_{LW}) and the length overall (L_{OA}). This drag can be added as a term in τ_d in Eqns. (34)-(36).

$$\tau_d = \frac{1}{2} \rho_{air} V_{rw}^2 \begin{bmatrix} C_X(\gamma_{rw}) A_{FW} \\ C_Y(\gamma_{rw}) A_{LW} \\ C_Z(\gamma_{rw}) A_{FW} L_{aa} \end{bmatrix} \quad (34)$$

$$V_{rw} = \sqrt{u_{rw}^2 + v_{rw}^2} \quad (35)$$

$$\gamma_{rw} = -atan2(v_{rw}, u_{rw}) \quad (36)$$

If the boat is symmetrical about the xz-plane [4] [11], the wind coefficients are defined by Eqns. (37)-(39). The coefficients are experimentally determined and are non-dimensional.

$$C_X(\gamma_{rw}) = -c_x \cos(\gamma_{rw}) \quad (37)$$

$$C_Y(\gamma_{rw}) = c_y \sin(\gamma_{rw}) \quad (38)$$

$$C_Z(\gamma_{rw}) = c_z \sin(2\gamma_{rw}) \quad (39)$$

2.3. Thruster Model

The quality of the thruster model shapes how well the boat model is expected to respond. The more accurate the thruster model in determining its thrust, the more accurate the motion and estimation results can become. Making a fully identified thruster model requires knowing the effects of the propeller's inertia and the water flow effect through the propeller which needs its own model identification and estimation. Simple thruster models are favored for simplicity and ease of estimation for steady maneuvers in calm waters to test the expected motion and limits of the vessel. Several papers take simple approaches more often in surface vehicles with Mask [3]

and more complex models used in more underwater or thruster identification testing with Häusler Häusler [12].

To get body forces (Eqn. (40)), the individual thrusters need to sum their forces and moments about the C.G. point. The C.G. is the x- and y-axis distance from the user-set Center of Origin (C.O.). The C.O. is a point on the vessel to reference all distances and dynamics from. Having any offset from the C.G. with a differentially thrusted boat will lead to a yaw moment generated due to surge motion. The T_x and T_y forces are simply summed with the moments being generated by their position from the C.G. The thruster's forces for the port (p) and starboard (s) sides can have an x- and y-direction.

$$\vec{\tau} = \{T_x T_y M_z\}^T \quad (40)$$

$$\vec{r} = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ -y_p & x_p & -y_s & x_s \end{bmatrix} \quad (41)$$

$$\vec{\tau}_T = \{T_{xp}, T_{yp}, T_{xs}, T_{ys}\}^T \quad (42)$$

$$\vec{\tau} = \vec{r} \times \vec{\tau}_T \quad (43)$$

2.3.1. Throttle Mapping

The simplest method of motor modeling is to have a known relationship of input to output. A commonly used method is a linear regression between the input throttle and the output forces/speeds.

The force is either based on datasheets from the manufacturer's specifications or related to command versus force measured from different throttle values. The relationship can be determined by a bollard test where the throttle command is related to the force measured by a force gauge that prevents motion. This model operates with the assumption of negligible change in waterflow such as no current effects and steady motion [13] [4]. Equations (44)-(45) define the relationship of

thrust based on the current thruster RPM percentage multiplied by the maximum attainable thrust from the manufacturer's specification.

$$T_p = T_{max} \left(\frac{\omega_p}{\omega_{max}} \right) \quad (44)$$

$$T_s = T_{max} \left(\frac{\omega_s}{\omega_{max}} \right) \quad (45)$$

where:

$$\omega - \text{RPM of the thruster} \left(\frac{rev}{s} \right)$$

2.3.2. Open-Water Motor Model

A commonly used model for marine vessels is the open-water motor model. This model generates the thrust and torque of a thruster using a coefficient known as the advance ratio (J_0). This model is used for keeping a minimum speed to the water with respect to control surfaces [12] and encompasses the regions of motion where the thrust is related to its direction of travel (Eqn. (46)-(48)).

$$J_0 = \frac{v_a}{\omega d} \quad (46)$$

$$T = \rho d^4 k_T (J_0) \omega^2 \quad (47)$$

$$Q = \rho d^5 k_Q (J_0) \omega^2 \quad (48)$$

where:

v_a – vessel's advance speed [m/s]

ω – propeller rotational speed [rad/s]

d – propeller diameter [m]

k_T/k_Q – coefficient of thrust/torque respectively

ρ – water density [kg/m^3]

T – estimated thrust [N]

Q – estimated torque [Nm]

J_0 – advance ratio

The issue with this model is that it is not suited for operations that require fast turns or sudden changes in direction. The forces modeled only consider the thrust effects acting with the speed of the vessel. This method does not capture reversing thrust in motion like “braking” or station-keeping accurately. To gather data that accounts for the changes in motion and more water effects in the system identification data, a more complex model needs to be used.

2.4. System Identification

2.4.1. Modeling Techniques

There are a few primary techniques used for modeling. The most common one is the white box model where the dynamics are explicitly related to the physics of the physical object [2] [14]. This method carries mathematical and physical assumptions that bring about a more predictable and valid solution. The downside of this technique is that the simulated system being modelled will not completely match the real system due to unknown physical effects that occur. Mathematical white box models were generated in Simulink by Benedict [15]. His models were focused on the drag equation of a surface moving through water based on commonly-used parameters. The overall model will provide insight into the motions, but it cannot capture all the expected coupling effects of the vessel that tend to occur between states such as surge and yaw rate coupling.

The black box model uses the empirical data collected from the real system to generate the model. The model presented by Fossen on the truncated-Taylor series is a form of fitting a multiple-in-multiple-out (MIMO) system model to a vessel. The model follows no direct physical representation, but it can still represent the vessel that is being tested.

The grey-box model takes part of both white- and black-box models to create a more specialized physical model of the system being used. The model becomes more descriptive of the actual vessel, in this case, while being defined by its white-box model form. The model follows

some form of white box modeling such as the rigid-body and second-order drag terms in the second-order model. The second-order model has terms that still need to be fit because no direct correlation exists to physical aspects of the model. For example, the cross term Y_{rv} is related to the sway force due to yaw rate but is fitted rather than using a direct, physically-related equation. In equation (49), the parameter is summed in the 3rd column to another quadratic term. This shows the effects the term can have on the sway force but is hard to directly measure since this drag term is in relation to the current sway speed and yaw rate.

$$D_{NL, Y}(\vec{v}) = \begin{bmatrix} \dots & & & & & \\ Y_{uu}|u| & Y_{vv}|v| + Y_{vr}|r| & Y_{rr}|r| + Y_{rv}|v| & & & \\ \dots & & & & & \\ & & & & & \dots \end{bmatrix} \quad (49)$$

$$\vec{v} = \{u, v, r\} \quad (50)$$

The grey-box is the preferred method to perform the system identification with work being attempted in by Mask where a Taylor-series model was fit based on parameters previously determined by a Fossen boat model [3]. The research by Mask was only done with Fossen's simulated boat model with guidelines on how to adapt their identification on their WAMV12 platform. This current research moves towards continuing this approach using the WAMV16 for identification based on Mask's initial work.

2.4.2. Optimization Techniques

Optimization is the process of finding the values of independent parameters that minimizes or maximizes a function [16]. There are two different classifications of optimizers: local and global. Local optimizers search for local minimum(s) in a function that fit the parameters for certain cases, but do not capture the overall minimum of the function. A global minimum is where the function value is smaller than or equal to all other feasible points. A local minimum in this case can get stuck in cases where a function converges on the first or more easily accepted minimum it finds.

Local optimization is beneficial for models that have an established function that needs a better fit. In [3], Mask uses the nonlinear grey-box model function in MATLAB to estimate and validate a Fossen boat model. The work develops a methodology for estimating parameters for vessels by scaling the known parameters by a ratio and then simulating them to obtain training and validation data sets. The optimizer optimizes the original model parameters and attempts to fit to the different scaled models. The nonlinear grey-box model by MathWorks is designed to search for a more optimal solution using sets of local optimization techniques such as nonlinear least squares and Quasi-Newton methods.

Global optimization is beneficial to searching a space that may have little to no initial parameters estimates for the search to reference. In this work, MATLAB is utilized to train data, thus the methods being explored are restricted to what algorithms MATLAB has implemented due to having robust algorithms and options to verify works. The optimizers work similarly to the local variants in terms that the algorithm search for the local minimum near it. The change is how the iterations of the optimizer change at each step.

One global optimizer is the genetic algorithm (GA) [17] [18], which searches functions using natural selection ideologies. The GA takes a population of individual solutions (i.e. parameters) and randomly assigns them as parents to produce new solutions for the next generation of the population. There are 3 main rules for the algorithm to expand this way. The selection rule selects the parents to contribute the children for the next generation. The crossover rule (Figure 3) combines 2 parents to form the next generation children. The mutation rule (Figure 4) applies random changes to the parent to form the next children. This evolves the population over time towards a global solution. The function that is being optimized is known as a fitness function. The fitness function describes how well the function fits and how the population can continue to evolve.

The function in this work is the error score calculated from the actual data versus the simulated results.

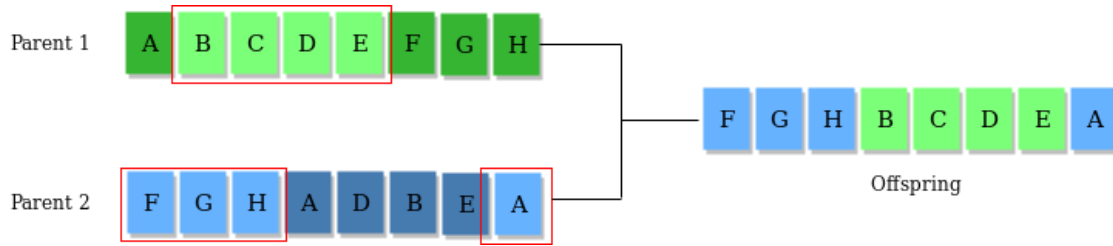


Figure 3: Example of using the Crossover Operator [19].



Figure 4: Example of using the Mutation operator [19].

Another type of global optimizer is the particle swarm (PS), which uses an animal swarm approach of moving in relation to the swarm by sharing its knowledge with each other. Every individual in the swarm is known as a particle. The particle moves along the search space given its own position and velocity and its goal represented by the swarm's overall position and its current search direction. This approach is also a pseudo-random search like the GA, but maintains its solutions based on the best of the swarm.

The GA and PS methods both require some sort of insight into the problem for some initial states to get a better set of parameters. The fitness of the function will only approach a solution when working with semi-unknown physical systems that may be over- or under-constrained but work with the current set of data sets. This is known as overfitting [20] where systems are over-parameterized to fixate on certain cases rather than being more flexible in other valid cases. Using a wide set of useful data sets for training the parameters, then using untouched data sets to validate their fit in similar and different conditions. The training and validation sets should cover the

expected motion in all cases that should be considered. Performing cross-validation is helpful to reduce overtraining by reducing the amount of data set collection by reusing some training as validation and those validation as training. A secondary check for determining the model's potential motion is to simulate basic motion cases like steady-state motion cases at fixed inputs.

2.4.3. System Identification Boat Maneuvers

There are sets of standards for identifying maneuvering capabilities and parameters for marine crafts. The standards have been commonly used in helping determine the unknown parameters for different identification techniques since the approaches have certain goals to them [21]. The required tests presented in the standards are the turning, zig-zag, and stopping tests.

The vessels presented in the standard are typically large ships to cargo ships. The general setup is some thruster configuration that is mostly concerned with solely forward thrust and a rudder-based turn. The WAMV16 used relies on differential thrust via aft mounted thrusters on the port and starboard sides. To make a comparison to other typical rudder-based ships, the USV will operate in 2 different states. First, any required forward motion will require both thrusters to operate at the same throttle level. Secondly, any rudder-deflections will be based on one of the thrusters having more power. For example, a zig-zag maneuver will have the WAMV drive forward to speed on both thrusters, then when turning, the thrust will alternate which thruster is active while the other will be idle. This motion is only considered for the sake of the standards and any other test is subject to additional thrust configurations as needed. The last note on the setup is that all of the motions performed are done through open-loop commands with the exception of any test checkpoints detected (i.e. reaching a target heading while zig-zagging to trigger the change in thrust to turn) will just command the WAMV to another open-loop command.

The stopping test (Figure 5) is known for determining the vessel's ability to stop from an initial velocity. In the literature [21], this required the ship to reach steady-state speed, then command a full-astern until the ship is sitting dead in the water.

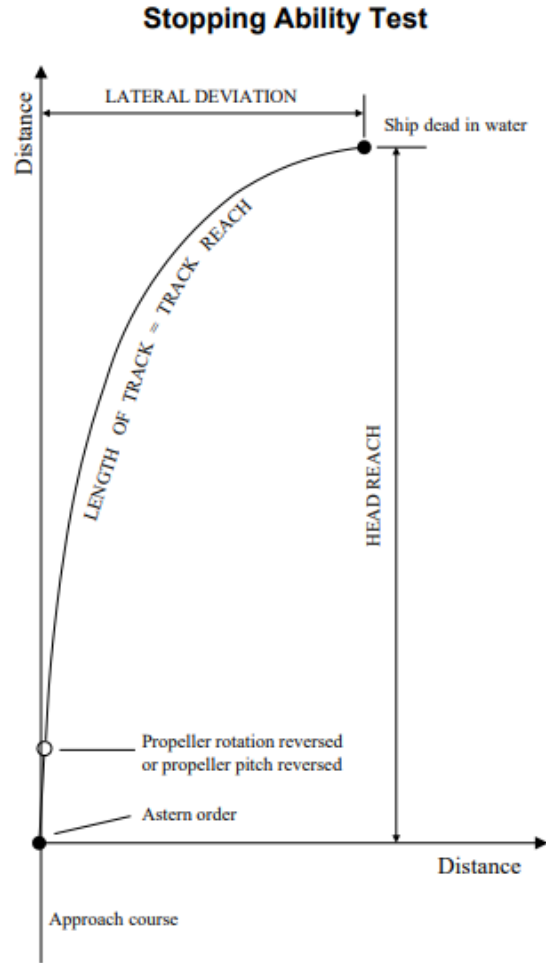


Figure 5: A diagram of the standard stopping maneuver [22].

The turning test [21] is designed to determine the ship's tactical diameter at different rudder angles given some cruise speed. This test begins to speed forward with no rudder action until a steady speed is achieved. Once the speed has been achieved, the rudder is set to a direction and held for at least 2 revolutions.

Turning Circle Test

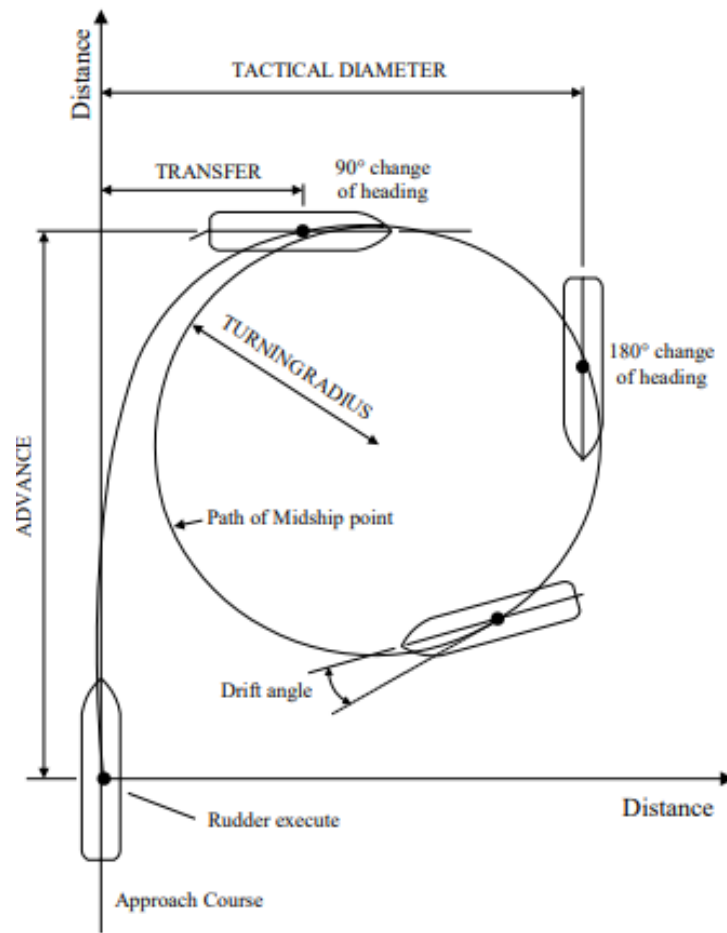


Figure 6: A diagram of the standard stopping maneuver [22].

The zigzag test [13] [21] (Figure 7) is designed to test vessel course-keeping and heading overshoot. The vessel starts moving forward at a constant heading, then the rudder is diverted to an angle and remains there until the vessel crosses that angle difference. Once the vessel changes heading, the rudder is commanded to the same heading change on the opposite side of the initial heading. Repeating this maneuver with different delta angles and speeds help determine the hydrodynamic coupling between the surge and yaw motions.

10/10 Zig-zag Maneuver Test

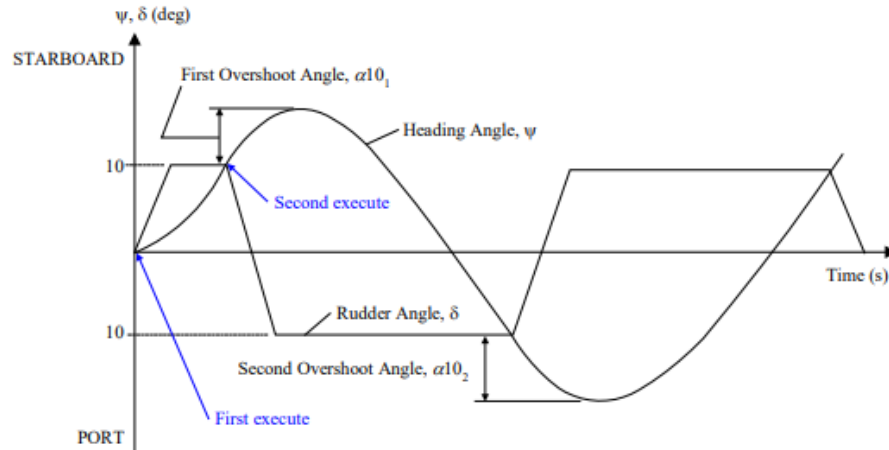


Figure 7: A diagram of the standard zig-zag maneuver [22].

With these tests giving expected, constant motion, the measured position drift can help isolate a factor of the average relative velocity of disturbances [21]. Using the turning test can determine that drift as shown in Figure 8.

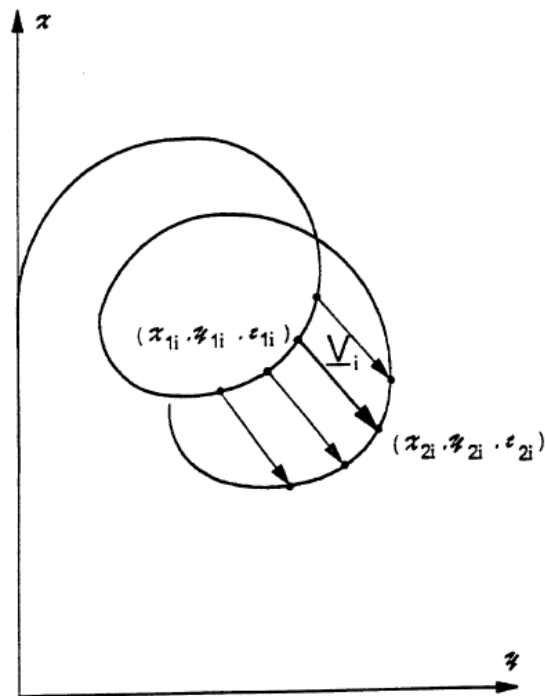


Figure 8: Determine the relative velocity of the water due to global disturbance [21]. The current vector defines the $\{x,y,t\}$ points to differentiate the average disturbance over time.

Chapter III

Methodology

3.1. Introduction

To generate the parameters for the WAMV, a set of standard maneuvers are used to train the dynamic model for use in simulation and controls using a GA optimizer. The process will require 3 phases: generating maneuvering data sets for training and validation, optimizing a system model using training sets of data, and analyzing results using validation data sets. These steps will be used to determine the coupled-state dynamics of a WAMV16 as defined by a Fossen model for marine vessels (Eqn. (51)). The maneuvers used for fitting will search for two variations of the Fossen model: sway-yaw coupling only, and full dynamic-state coupling. The differences between these model attempts are that the former has no significant coupling parameters to surge besides the Coriolis effect, and the latter will have more parameters to estimate.

$$(M_{RB} + M_{AM})\dot{\vec{v}}_r + (C_{AM}(\vec{v}_r) + C_{RB}(\vec{v}_r))\vec{v}_r + D(\vec{v}_r)\vec{v}_r = \vec{\tau} + \vec{\tau}_d \quad (51)$$

3.2. Limitations and Assumptions

This work will model a WAMV16 using a 2nd order maneuvering model as defined by Fossen. The model is limited to defining a 3-DOF motion of surge velocity, sway velocity, and yaw rate neglecting any motion due to waves and any other rigid-body motions, such as heave velocity and roll and pitch rates. The rigid-body motions are defined at the C.O. of the vessel which is placed at the C.G. of the vessel. Due to the model being nonlinear with 2nd order traits, the model operates best with positive forward speeds and the state ranges based on the parabolic drag limits. The limitations of the model can be suppressed by the maximum inputs forces that are used for training.

The control inputs of the model are linearized thrust and wind speeds. The thruster model assumes that the thrust estimated is linearly proportional to the thruster's RPM. The thrusters will

ignore the effects of water flow and torque for thrust estimation due to lack of equipment to fully identify the effects.

The wind model is a basic quadratic drag equation based on the angle of attack of the exposed surface area to the wind. The measured wind velocities at each time step are used as a direct model input. This model also assumes the surface area of the vessel is symmetrical.

3.3. Initial Parameter Estimation

This section will determine the methods used to obtain the basic physical insight of the WAMV's dynamics. Physical insight is where the dynamic parameters can be referenced to its direct input and motion relation. An example of physical insight is the surge drag characteristics of the WAMV because the simplified drag of water is a quadratic term and two of the surge parameters are a quadratic equation pair. The initial parameters that can be collected are the rigid-body terms and a few hydrodynamic drag terms. All the available parameters for the model are presented and given dimensional units in Table 1 for physical reference.

Table 1: Parameters set to be estimated and their units of the 2nd-order Fossen model.

Rigid-Body		X (Surge)		Y (Sway)		N (Yaw)	
$mass$	kg	$X_{\dot{u}}$	kg	$Y_{\dot{u}}$	kg	$N_{\dot{u}}$	kgm
I_{zz}	kgm^2	X_u	$\frac{kg}{s}$	Y_u	$\frac{kg}{s}$	N_u	$\frac{kgm}{s}$
x_{CG}	m	X_{uu}	$\frac{kg}{m}$	Y_{uu}	$\frac{kg}{m}$	N_{uu}	kg
y_{CG}	m						
Wind		$X_{\dot{v}}$	kg	$Y_{\dot{v}}$	kg	$N_{\dot{v}}$	kgm
		X_v	$\frac{kg}{s}$	Y_v	$\frac{kg}{s}$	N_v	$\frac{kgm}{s}$
A_{FW}	m^2	X_{vv}	$\frac{kg}{m}$	Y_{vv}	kgm	N_{vv}	kg
A_{LW}	m^2			Y_{vr}	kg	N_{vr}	kgm
L_{aa}	m	$X_{\dot{r}}$	kgm	$Y_{\dot{r}}$	kgm	$N_{\dot{r}}$	kgm^2
c_x	N.D.	X_r	$\frac{kgm}{s}$	Y_r	$\frac{kgm}{s}$	N_r	$\frac{kgm^2}{s}$
c_y	N.D.	X_{rr}	kgm	Y_{rr}	kgm	N_{rr}	kgm^2
c_z	N.D.			Y_{rv}	kg	N_{rv}	kgm

3.3.1. Rigid-Body

The initial rigid-body parameters were determined through the Solidworks CAD model of the WAMV (Figure 9). The model parameters required are the mass, I_{zz} , and the C.G. location from the WAMV's C.O., which will be displayed in Table 2. The C.O. (Figure 9) is an arbitrary point on the vessel to reference inputs and dynamic motions from. The variables

$mass$, I_{zz} , x_{CG} , and y_{CG} , were estimated from the CAD model and shown in Table 2. With these parameters, the rigid-body Mass and Coriolis terms are expressed in Eqns. (52)-(53).

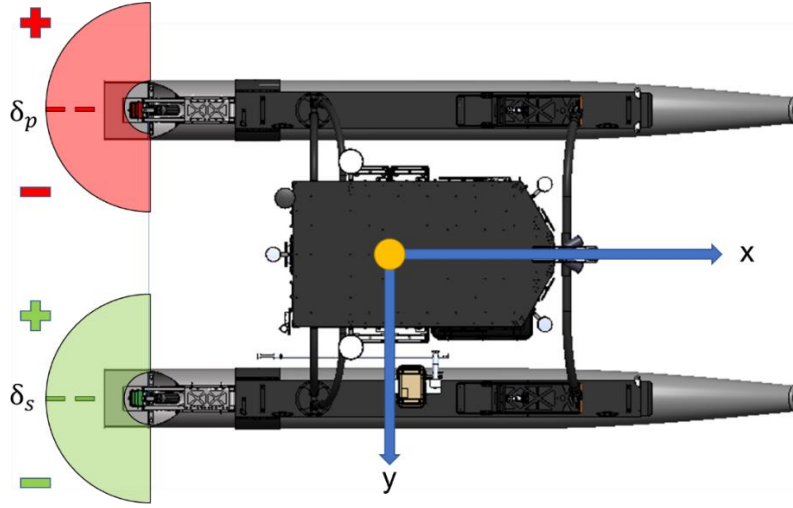


Figure 9: Top view of the Minion WAMV16 with C.O. (yellow circle) and azimuth angle (δ) ranges for port and starboard displayed.

Table 2: The initial rigid-body parameters used in the estimation.

Rigid-Body	
$mass$	348.39
I_{zz}	525.39
x_{CG}	0
y_{CG}	0

$$M_{RB} = \begin{bmatrix} m & 0 & -my_{CG} \\ 0 & m & mx_{CG} \\ -my_{CG} & mx_{CG} & I_{zz} \end{bmatrix} \quad (52)$$

$$C_{RB} = \begin{bmatrix} 0 & 0 & -m(x_{CG}r + v) \\ 0 & 0 & -m(y_{CG}r - u) \\ m(x_{CG}r + v) & m(y_{CG}r - u) & 0 \end{bmatrix} \quad (53)$$

3.3.2. Hydrodynamic

The hydrodynamic parameters are defined by the effects of the WAMV in the water due to drag and damping. To determine the basic, initial parameters, the sets of tests needed are steady-state drag tests for the controllable motions available: surge and yaw rate. From the full set of

Coriolis (Eqn. (54)) and drag terms defined in Eqns. (55)-(56), the primarily determined parameters will be X_u , X_{uu} , N_r , and N_{rr} . These terms are the linear-quadratic pair of coefficients for drag motion of surge and yaw respectively. All test data sets will be presented graphically in Appendix 1.

$$C_{AM}(\vec{v}) = \begin{bmatrix} 0 & 0 & Y_{\dot{u}}u + Y_{\dot{v}}v + Y_{\dot{r}}r \\ 0 & 0 & -X_{\dot{u}}u - X_{\dot{v}}v - X_{\dot{r}}r \\ -Y_{\dot{u}}u - Y_{\dot{v}}v - Y_{\dot{r}}r & X_{\dot{u}}u + X_{\dot{v}}v + X_{\dot{r}}r & 0 \end{bmatrix} \quad (54)$$

$$D_L = \begin{bmatrix} X_u & X_v & X_r \\ Y_u & Y_v & Y_r \\ N_u & N_v & N_r \end{bmatrix} \quad (55)$$

$$D_{NL}(\vec{v}) = \begin{bmatrix} X_{uu}|u| & X_{vv}|v| & X_{rr}|r| \\ Y_{uu}|u| & Y_{vv}|v| + Y_{vr}|r| & Y_{rr}|r| + Y_{rv}|v| \\ N_{uu}|u| & N_{vv}|v| + N_{vr}|r| & N_{rr}|r| + N_{rv}|v| \end{bmatrix} \quad (56)$$

3.3.3. Input Model Estimation

3.3.3.1. Thruster Model Estimation

The input method used for the differential thruster setup is the throttle mapping method. The model maps the thrust of the thruster proportionally to its RPM (Eqns. (44)-(45)). This basic estimation of the motor model is used due to a lack of equipment to estimate an accurate model. This model assumes that the effects of water flow through the propeller and thruster torque estimates are ignored. The thrusters used in this work are asymmetrical, meaning when the thruster commands reverse, the linear scaling is based on T_{min} in place of T_{max} as shown in

Table 3. The thrusters map is displayed in Figure 10 with the extreme thruster speeds are at -2300 and 2300 RPM.

$$T_p = T_{max} \left(\frac{\omega_p}{\omega_{max}} \right) \quad (57)$$

$$T_s = T_{max} \left(\frac{\omega_s}{\omega_{max}} \right) \quad (58)$$

Table 3: The thruster ranges used to scale input forces/moment.

Thruster Force @ 2300 RPM	
T_{max}	600
T_{min}	400

Port Position (x, y) at C. G.	$(-1.71, -1)$
Star. Position (x, y) at C. G.	$(-1.71, 1)$

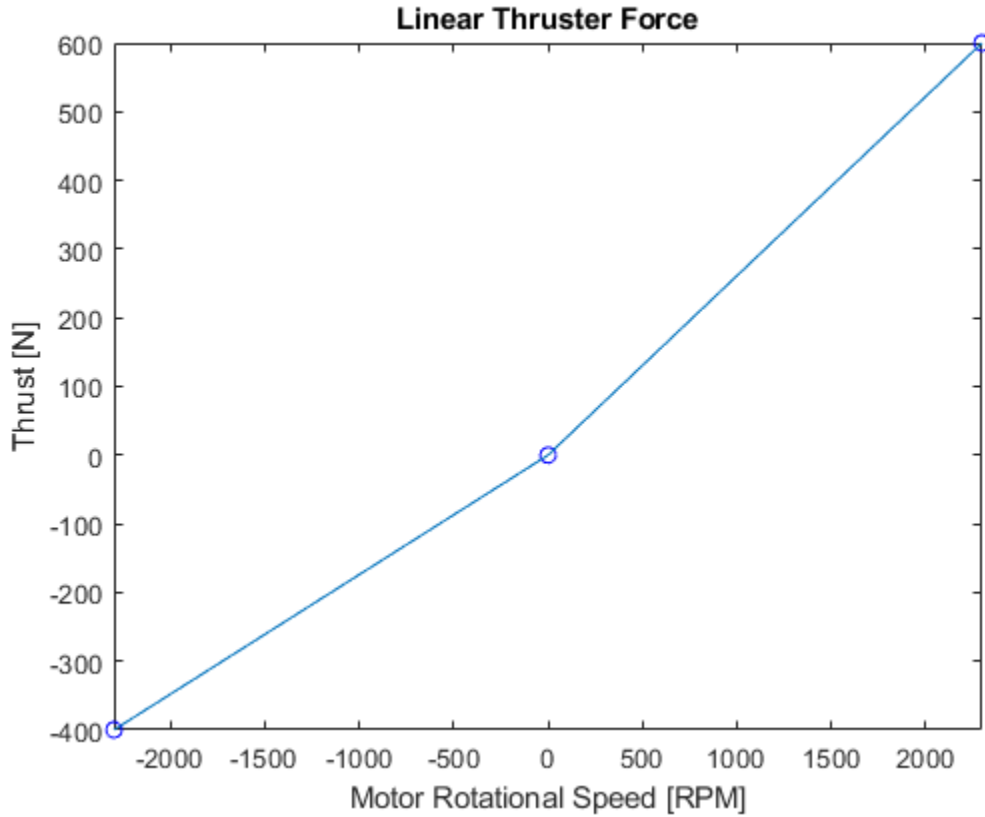


Figure 10: Linear thrust curve for asymmetrical thrusters.

3.3.3.2. Wind Model Estimation

The wind model used is to estimate air drag on the WAMV given a symmetrical assumption. The input measurement is the relative wind velocities (u_{rw}, v_{rw}) that the WAMV experienced at each time step. The velocity and heading of the simulated model will be used to reference the relative velocity in Eqns. (60)-(61) to calculate the air drag in Eqn. (59).

$$\tau_d = \frac{1}{2} \rho_{air} V_{rw}^2 \begin{bmatrix} C_X(\gamma_{rw}) A_{FW} \\ C_Y(\gamma_{rw}) A_{LW} \\ C_Z(\gamma_{rw}) A_{FW} L_{aa} \end{bmatrix} \quad (59)$$

$$V_{rw} = \sqrt{u_{rw}^2 + v_{rw}^2} \quad (60)$$

$$\gamma_{rw} = -atan2(v_{rw}, u_{rw}) \quad (61)$$

3.3.4. Maneuvering Testing

The maneuvering data sets discussed in this section will be used to create the training and validation data sets to be optimized. The maneuvers are divided into three sections: drift due to disturbance, steady-state, and the standard maneuvering data sets.

3.3.4.1. Drift Test

The drift test captures the external disturbances that affect the system (Tests displayed in Appendix 1.1). The WAMV is set to move with the disturbances as the only input to the system. This location for the test is done in a lake where the dominant disturbance force is the wind. This test is done throughout the test day, usually every hour to check for average change in disturbances and to have multiple training sets for the disturbance models. All drift tests were done with the vessel facing aft towards the wind to characterize surge drag effects.

The coefficients determined from the drift test are set the rigid-body and wind-based coefficients. The rigid-body coefficients are defined by the surface areas and length of the vessel. A_{FW} , A_{LW} , and L_{aa} are the surface areas of the front and side of the vessel and the length of the vessel respectively. These are estimated based on the sum of the rectangle and cylindrical areas that are exposed to the air. The force/moments coefficients (c_x , c_y , and c_z), in Eqns. (62)-(64), are the parameters that correlate the relative wind speed dynamic effects that act on the vessel (

Table 4).

$$C_X(\gamma_{rw}) = -c_x \cos(\gamma_{rw}) \quad (62)$$

$$C_Y(\gamma_{rw}) = c_y \sin(\gamma_{rw}) \quad (63)$$

$$C_Z(\gamma_{rw}) = c_z \sin(2\gamma_{rw}) \quad (64)$$

Table 4: List of the wind parameters needed to estimate wind diforces and moments.

Wind Coefficients			
Rigid-Body		Force	
A_{FW}	81.45	c_x	estimated
A_{LW}	702.90	c_y	estimated
L_{aa}	4.55	c_z	estimated

3.3.4.2. Steady-State Tests

The steady-state tests capture the quadratic drag dynamics of the system. The states being tested are the steady-state drag dynamics with respect to the surge and yaw rate states. The tests command sets of constant throttle commands to achieve a steady-state motion and then the state is captured and tabulated to generate a quadratic polynomial fit. Table 5 shows the throttle settings used to gather the drag data sets. The surge test was performed moving forward and the yaw drag was tested in both rotational directions. The WAMV will be rotating in both directions to account for drag fits that help capture the yaw drag's symmetry.

Table 5: Throttle settings used for the surge and yaw drag tests.

Surge Drag Throttle Settings		Yaw Drag Throttle Settings	
% Port	% Starboard	% Port	% Starboard
10	10	-10	10
20	20	-20	20
30	30	-30	30
40	40	-40	40
50	50	-50	50
60	60	-60	60
70	70	-70	70
80	80	-80	80
90	90	-90	90
100	100	-100	100

3.3.4.3. Stopping Test

The stopping test is altered from the standard stopping test in order designed to capture the natural hydrodynamic drag and damping (Eqns. (65)-(67)) from the surge state (Appendix 1.2).

The WAMV will move with the aft facing the wind for each test. The stopping test sets both thrusters to the same throttle command until a steady-state surge speed is achieved. Then after 5 seconds of holding constant speed, the throttle settings are set to 0% and the WAMV drifts to either a stop or steady-state speed due to disturbance. The 50% throttle test inputs are displayed in Figure 11.

$$M_{AM} = \begin{bmatrix} \mathbf{X}_u & X_v & X_r \\ Y_u & Y_v & Y_r \\ N_u & N_v & N_r \end{bmatrix} \quad (65)$$

$$D_L = \begin{bmatrix} \mathbf{X}_u & X_v & X_r \\ Y_u & Y_v & Y_r \\ N_u & N_v & N_r \end{bmatrix} \quad (66)$$

$$D_{NL}(\vec{v}) = \begin{bmatrix} \mathbf{X}_{uu}|u| & X_{vv}|v| & X_{rr}|r| \\ Y_{uu}|u| & Y_{vv}|v| + Y_{vr}|r| & Y_{rr}|r| + Y_{rv}|v| \\ N_{uu}|u| & N_{vv}|v| + N_{vr}|r| & N_{rr}|r| + N_{rv}|v| \end{bmatrix} \quad (67)$$

Table 6: Throttle settings used for the stopping tests.

% Port	% Starboard
10	10
20	20
30	30
40	40
50	50
60	60
70	70
80	80
90	90
100	100

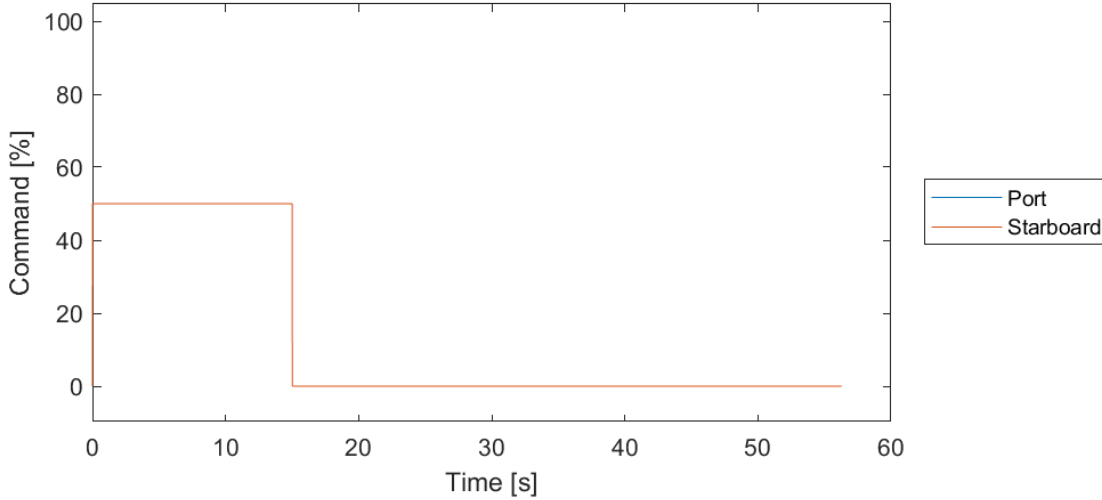


Figure 11: Input commands for a 50% stopping test. Both port and starboard are commanded simultaneously.

3.3.4.4. Turning Test

The turning test (Appendix 1.4) determines the amount of basic drag coupling and damping between the sway-yaw states and potential surge states (Eqns. (68)-(70)). The WAMV drives with the disturbances until a steady state speed is reached, then one of the thrusters is set to remain active at the same throttle level used to hold speed while the other is stopped (

Table 7). The test lasts for at least two full revolutions to visually capture the direction of the disturbances overall. The 70% turning test inputs are displayed in Figure 12.

$$C_{AM}(\vec{v}) = \begin{bmatrix} 0 & 0 & Y_{\dot{u}}u + Y_{\dot{v}}v + Y_{\dot{r}}r \\ 0 & 0 & -X_{\dot{u}}u - X_{\dot{v}}v - X_{\dot{r}}r \\ -Y_{\dot{u}}u - Y_{\dot{v}}v - Y_{\dot{r}}r & X_{\dot{u}}u + X_{\dot{v}}v + X_{\dot{r}}r & 0 \end{bmatrix} \quad (68)$$

$$D_L = \begin{bmatrix} X_u & X_v & X_r \\ Y_u & Y_v & Y_r \\ N_u & N_v & N_r \end{bmatrix} \quad (69)$$

$$D_{NL}(\vec{v}) = \begin{bmatrix} X_{uu}|u| & X_{vv}|v| & X_{rr}|r| \\ Y_{uu}|u| & Y_{vv}|v| + Y_{vr}|r| & Y_{rr}|r| + Y_{rv}|v| \\ N_{uu}|u| & N_{vv}|v| + N_{vr}|r| & N_{rr}|r| + N_{rv}|v| \end{bmatrix} \quad (70)$$

Table 7: Throttle settings used for the turning tests.

Surge Drag Throttle Settings		Yaw Drag Throttle Settings	
% Port	% Starboard	% Port	% Starboard
50	50	50	0
60	60	60	0
70	70	70	0
80	80	80	0
90	90	90	0
100	100	100	0
50	50	0	50
60	60	0	60
70	70	0	70
80	80	0	80
90	90	0	90
100	100	0	100

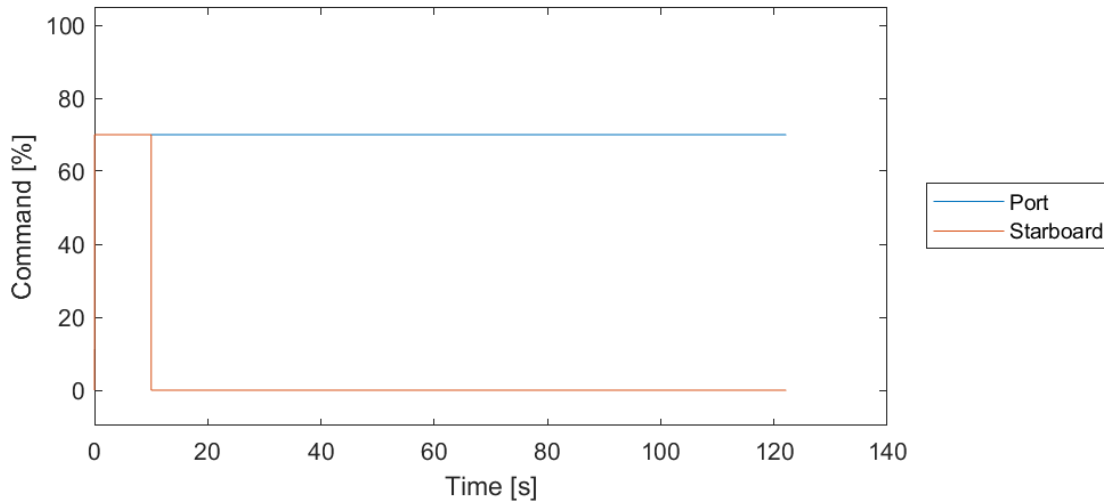


Figure 12: Input commands for a 70% turning test. The port thruster is commanded to stay on throughout the test.

3.3.4.5. Zig-zag Test

The zig-zag test (Appendix 1.5) determines similar parameters to the turning test but show more of the dynamic change in yaw rate based on the input response. This test starts with the same throttle commands to get to a steady surge speed. When the WAMV is ready to turn, the current heading angle is retained and used to determine when delta angles have occurred. If the starting heading was 30 deg in a 10-10 zig-zag, then the WAMV would change commands at 20 deg and 40 deg. This makes the heading the closed loop control and the throttle commands being open-

loop. Table 1 shows the commands used for each zig-zag test with the angle deltas tested for. The 100% 10-10 zig-zag inputs are displayed in Figure 13.

Table 8: Throttle settings used for the zig-zag tests.

% Motor Commands		Angles Used
50	0	10-10
60	0	20-20
70	0	35-35
80	0	
90	0	
100	0	

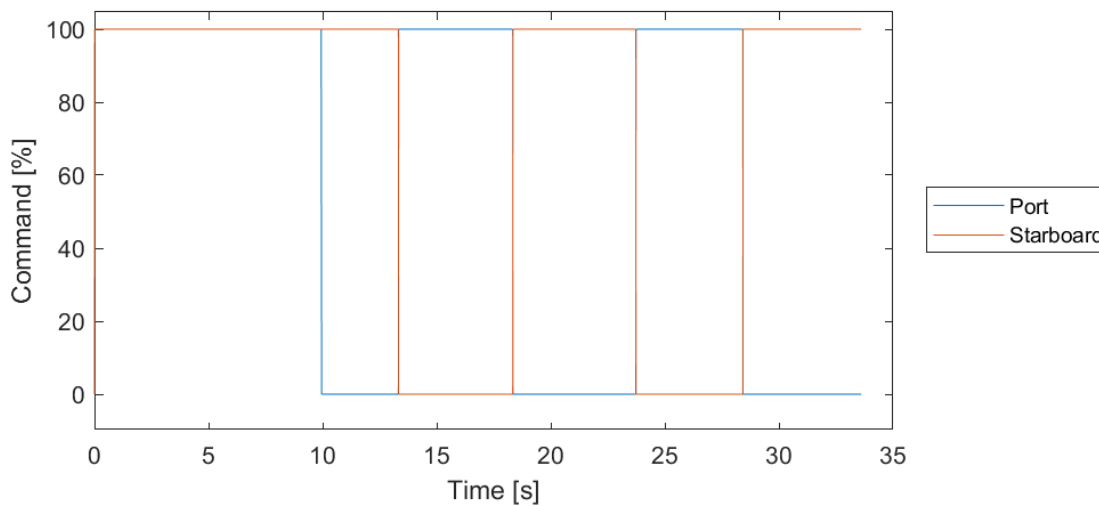


Figure 13: Input commands for a 100% turning test. The thruster commands oppose each other every 10 deg change from the initial heading angle throughout the test.

3.3.4.6. Figure-Eight Test

The figure-eight test (Appendix 1.6) validates the changes between the steady-state and dynamic sway-yaw relations. These tests are the turning tests that are set to change directions after every revolution. This set is used for validating the system model to handle how well the disturbance model pushes the WAMV including the drag from switching directions between the sway-yaw states. Table 9 shows the throttle tests used to generate a figure-eight resulting like Figure 14.

Table 9: Throttle settings used for the figure-eight tests.

Surge Drag Throttle Settings	
% Port	% Starboard
75	0
80	0

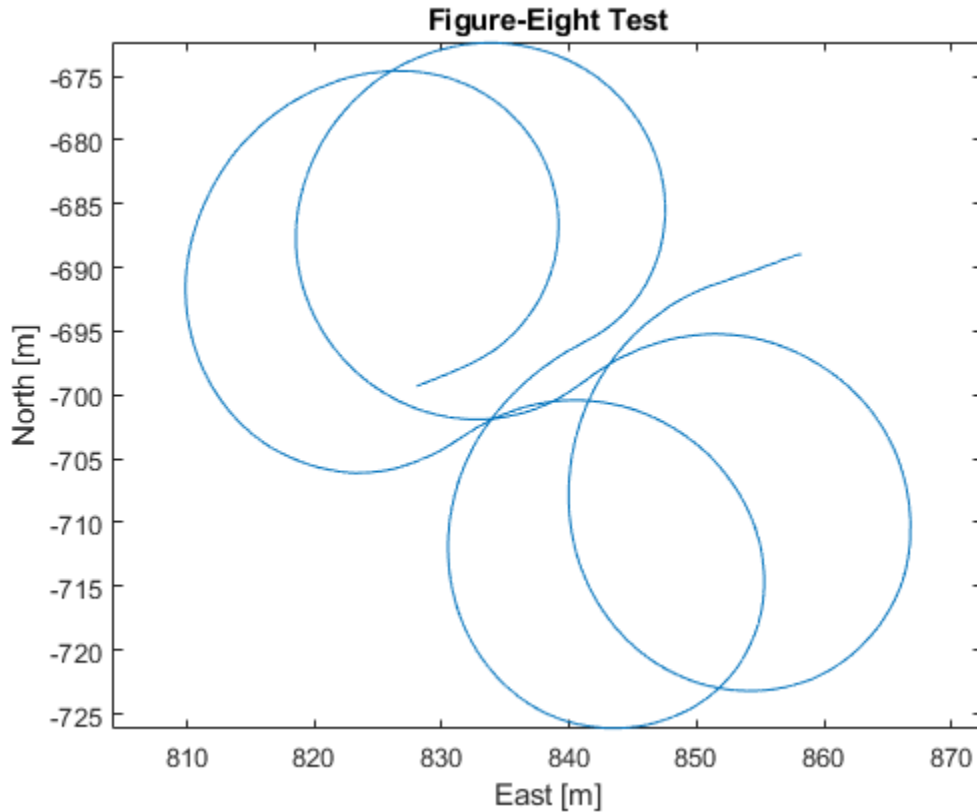


Figure 14: A figure-eight test taken at Lake Ashby with noticeable drift due to wind.

3.4. Genetic Algorithm Optimizer Setup

The optimizer being used is MATLAB's implementation of a genetic algorithm which stochastically evolves the population of parameters per iteration. The setup of the optimizer is in three steps: data sets insertion, the GA settings for optimization, and the error function for the model. Once all three steps are done, the validation is used to check the estimation results.

3.4.1. Genetic Algorithm Data Sets

The data sets used for training are set to be run in series through each optimization attempt. The logs are run with two phases: the disturbance data sets then the maneuver data sets. The disturbance data sets are run first through the optimizer in any order. The maneuvering tests are input in a random order to reduce the fitting towards one type of data set. Estimating the stopping test data sets may lead to strong surge drag results but may set bad yaw dynamics with the low fitness values. The optimizer may hit stopping limits, such as max generations and time limits, if poorly initialized that may not reach solutions that capture all the states visibly well. The first estimation will be initialized by the steady-state drag parameters for the surge and yaw rate states.

3.4.2. Genetic Algorithm Settings

The settings in Table 10 are for the GA optimizer function with the other settings not mentioned were left to the MATLAB 2018b defaults. The optimizer optimizes a population of 250 sets of parameters per generation. The maximum generation is the total amount of evolutions of the population of parameters that occur until the evolution stalls does not improve past the default function tolerance between generations.

Table 10: the GA settings used with MATLAB 2018b.

GA Settings	
Time Limit	6 hrs.
Max. Generations	500
Stall Generations	75
Population Size	250
Function Tolerance	1.0E-6

3.4.3. Genetic Algorithm Error Function

The error function of the optimizer generates the error value that the GA will minimize. The function takes in a data set of time, states, and inputs to compare to the simulated model. The 2nd-order Fossen model is simulated using MATLAB's ode45 solver given the current estimation

of the optimizer's system parameters. The error calculated is the sum of weighted squared errors of the six available states. The squared error is used to minimize the error function towards 0 and to increase the effect of error by scaling all the states in relation to one another.

$$error = \sum \left((state_{log} - state_{sim})^2 * weights \right) \quad (71)$$

The weights are set to correlate the errors between different dynamic values. The error of 0.1 m/s in linear speed is different than a 0.1 rad/s error in rotational speed, especially when the WAMV is observed to move at speeds up to 3.5 m/s at full thrust and ~0.5 rad/s in a full zero-radius turn. Table 11 has the list of weights assigned to each type of data set. The weights were experimentally determined through trial and error for how much each error should be scaled. The first three values are the surge speed, sway speed, and yaw rate. The last three pose states are the North position, East position, and Yaw angle. The pose states are there to act as another type of error bound that attempts to correct for integration errors in yaw angle most of the time. The drift test uses some error in the positions to help estimate disturbance inputs.

Table 11: The weights used for the GA optimizer for each test's data set.

Weights	
Drift	{1, 1, 3, 0.5, 0.5, 3}
Steady-State	{1, 1, 3, 0, 0, 3}
Stopping	{1, 1, 3, 0, 0, 1.5}
Turning	{1, 1, 3, 0, 0, 3}
Zig-Zag	{1, 1, 3, 0, 0, 2.5}
Figure-Eight	{1, 1, 3, 0, 0, 3}

3.4.4. Genetic Algorithm Validation

When the optimizer produces the final parameters, then validation simulations are performed by simulating the validation data sets with the new parameters. The validation states will show the visible fits of the states to show if the simulation looks appropriate and an error over the log will be calculated to determine the mean error of each individual state (Eqn. (71)).

$$\vec{v}_{error} = \sqrt{\frac{(state_{log} - state_{sim})^2}{\# total samples}} \quad (72)$$

If the resulting states do not meet expectations in some or all states, the estimation can be rerun with a supervised set of manually-tuned parameters. If a state is underestimated, the drag is too strong for the input. The drag may be related to its respective state or a coupled-state term that may have a stronger effect in certain state cases. The parameters that are edited and rerun to fit a general set of validation logs can help the optimizer minimize the error better in the next fit. When the parameters are initialized, change the parameter's upper and lower limits to allow the solution to search for more options or attempt to maintain certain state dynamics that appear favorable.

3.5. Equipment Used

3.5.1. USV

The vessel used is the Minion Unmanned Surface Vessel (USV) which is a WAMV16. The WAMV is a twin-hulled vessel that is designed to have the pontoons articulate independently to produce a steady deck motion in the presence of waves. The overall parameters of the WAMV is shown in Table 2, 4, and 12.

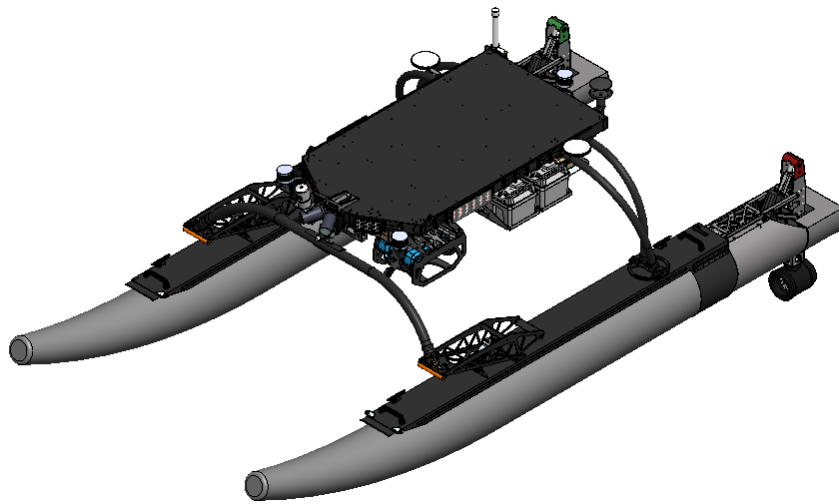


Figure 15: Picture of the Minion USV.

Table 12: Minion rigid-body parameters.

WAMV Parameters	
Length	4.8
Length of Waterline	4.00
Beam Width	2.4
Beam Width (Centers)	2.0

3.5.2. Propulsion

The USV's propulsion is equipped with a dual-thruster azimuthing propulsion system. The propulsion configuration comprises of 2 rim-driven propellers (Copenhagen Subsea VM Asymmetric thrusters, Figure 17) that are located at the rear end of the USV (one per side). The thrusters can produce approximately 600N of thrust forward in the Copenhagen static tests (Figure 16). The thrusters produce asymmetrical thrust that has 60% of it forward thrust being the maximum thrust in reverse. A Piktroniks motor controller is used to command and receive feedback at 100 ± 2 Hz with feedback from our custom motor interface board coming in at a steady 67 Hz. The feedback of the thruster primarily includes revolution per minute (RPM), the back electromotive force (emf), and motor current to log simultaneously with the localization system and generate the estimated forces and torques from each individual motor. The motor data will be kept constant per localization time step until the next motor update for logging purposes.

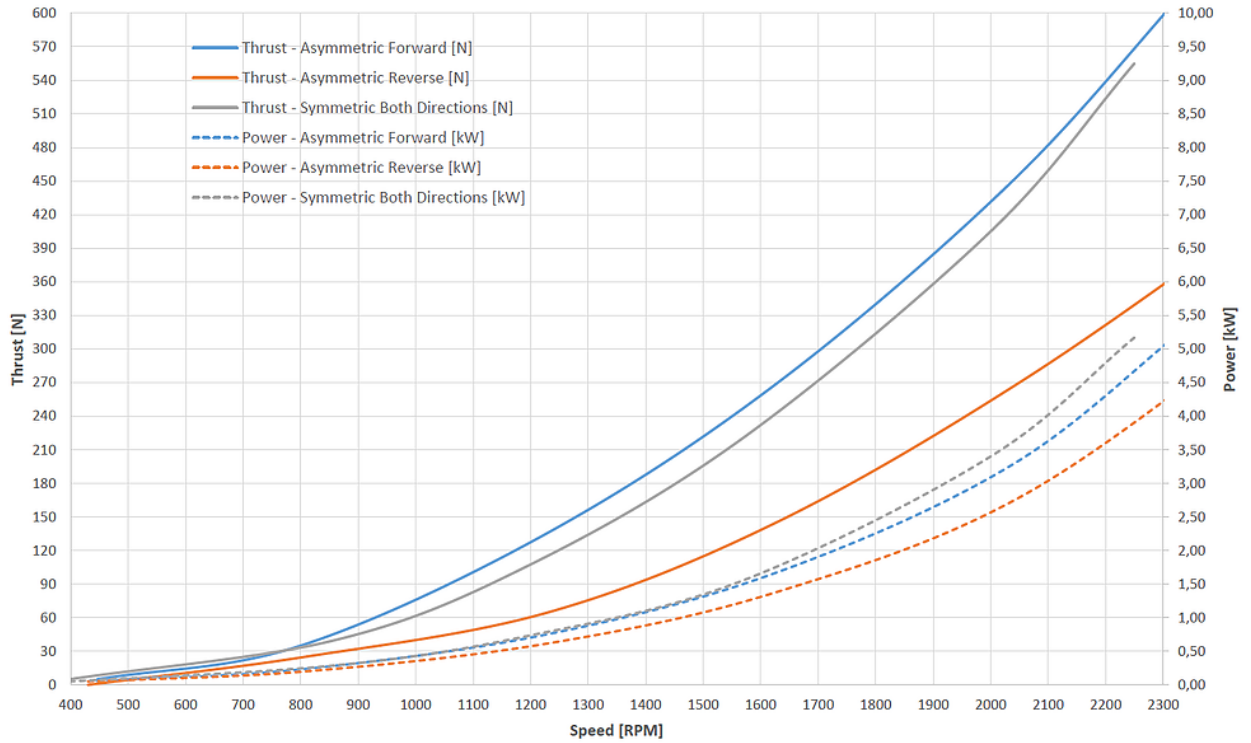


Figure 16: Performance Data for the Copenhagen VM from Copenhagen's data set at a hydrostatic state. The curves of interest are the "Thrust – Asymmetric Forward and Reverse".

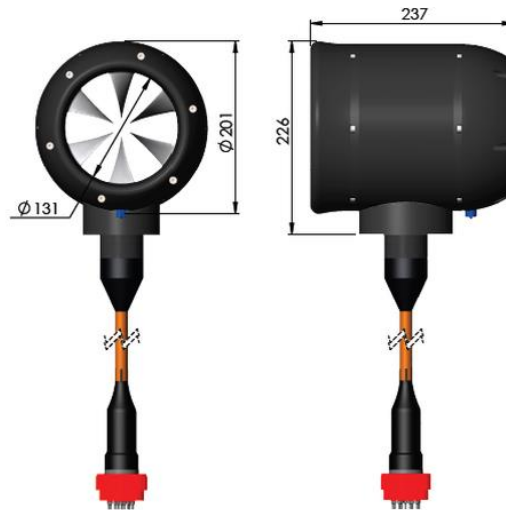


Figure 17: Copenhagen Subsea VM Asymmetric thrusters.

3.5.3. Localization

The localization suite consists of a TORC Pinpoint [23] and a WindSonic anemometer (Figure 18). The Pinpoint produces a fully-localized space in the NED frame using a dual-GPS antenna setup with a couple of 6-axis IMUs built-in. The Pinpoint was designed with the intention

and compatibility to use on automotive ground vehicles. That may lead to some errors in accurate sway data over very small speeds due to the assumption of predominantly moving in the x-axis. The Pinpoint generates body-frame velocities including surge, sway, and yaw-rate with errors of position within 10 cm, speed within 2 cm/s and roll, pitch, and yaw angles $\{0.0018, 0.0018, 0.0027\}$ rad ($\{0.1, 0.1, 0.17\}$ deg) [23]. The data fitting is noted to fit around the mean of the state with error since the report of the Pinpoint utilizes a nonlinear Kalman Filter that has a zero-mean Gaussian noise. The data rate is 100 ± 2 Hz to capture the state of the USV with the offsets of the Pinpoint set to the location of the origin of the USV. The WindSonic anemometer is a wind sensor that captures wind speeds relative to its orientation using an array of 4 ultrasonic sensors. The sensor can measure wind speeds at 4 Hz at all yaw angles up to a maximum of 60 m/s with 0.25 m/s accuracy error noted at 12 m/s in the user manual [24].



Figure 18: TORC Pinpoint GPS/INS (left) and WindSonic Anemometer (right).

3.6. Testing Area

The data was collected from a lake to identify wind as the primary disturbance. The test area was Lake Ashby in New Smyrna, FL where the average wind speeds were around 9 knots (4.6 m/s) with peaks up to 18 knots (9.2 m/s). This location was used for identification to estimate the wind along with system model. Also, by testing in a lake, the current does not exist, making estimation more straightforward.



Figure 19: Google view of the test area, Lake Ashby.

Chapter IV

Results

4.1. Overview

The results presented will cover the tests for the optimization results. The data will be presented in the order of disturbances, steady-state curve fitting, the sway-yaw coupling, and the full-dynamic coupling. The sway-yaw coupling has gone through three optimization attempts and the results of the full coupling has undergone one optimization. An analysis for using a linearized version of the model will be compared to the best fit model in this estimation set.

4.2. Disturbances

At the start of the test, the USV was left to drift with the disturbances. At Lake Ashby, the anemometer picked up an average wind speed around 4 m/s throughout the day and was used in determining the effects on the USV. The USV travelled around 0.5 m/s heading westward as show in Figure 20. These data sets are the first to run with the stead-state parameters to make estimates on the wind forces and attempt to make vessel parameter estimations.

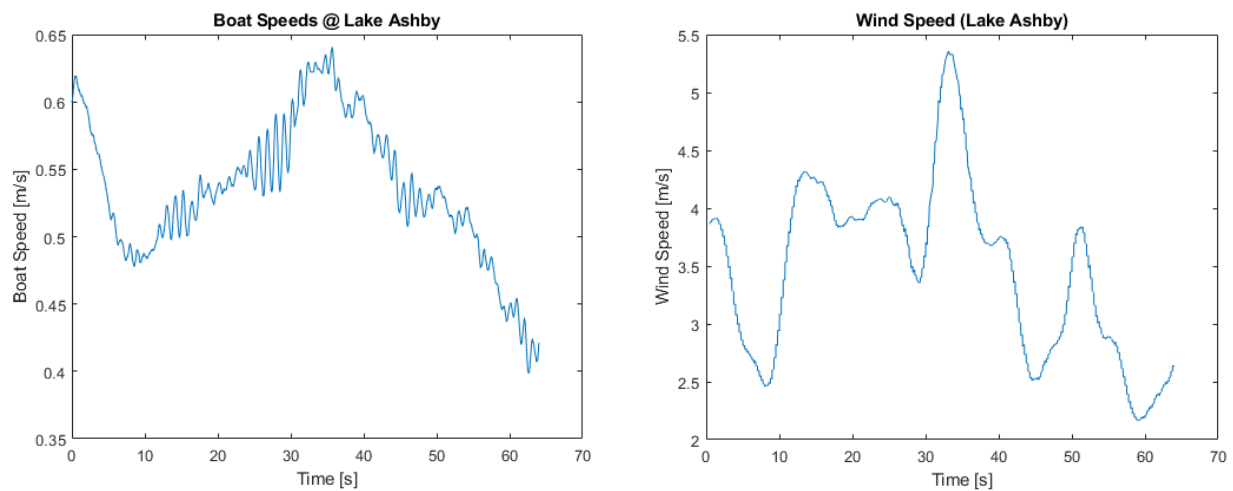


Figure 20: General wind disturbance at Lake Ashby. The overall USV speed was 0.5 m/s in 4 m/s winds.

4.3. Initial Parameter Estimation Results

4.3.1. Initial Surge Polynomial Fit

The surge-drag test was performed gathering the steady-state speeds from the stopping tests (Appendix 1.2). The speeds were collected before the throttle was commanded to 0%, and then tabulated into Excel. The resulting curve fit are shown in Figure 21 with the surge drag estimates shown in Table 13. The drag shows a linear fit, potentially due to using a linear motor model to estimate the thrust per data set. These estimates are set as part of the initial parameter population for the GA optimizer.

Table 13: The surge-drag results used for initial estimation.

Surge Parameters	
X_u	490.5
X_{uu}	-25.88

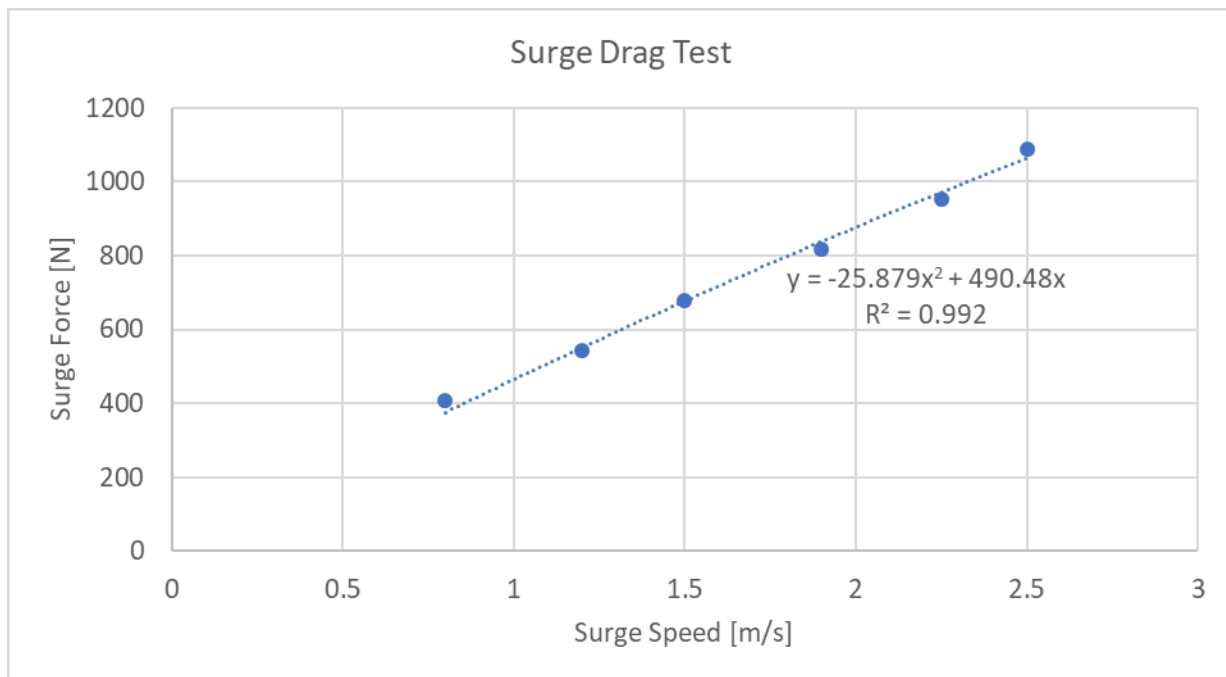


Figure 21: Surge drag curve-fit from the steady-state test ranging from 0.75 to 2.5 m/s.

4.3.2. Initial Yaw Rate Polynomial Fit

The yaw-drag test was collected through steady-state yaw rate rotations with the results shown in Figure 22. The state for the yaw rate did not show unique changes until around 30 seconds into the shown data set. The inputs before that time were set up to 30% with the yaw rate showing an average change around -0.06 rad/s which caused a separation in tabulated yaw rates. The points that were collected are still valid states and are used for an initial estimate for the yaw drag parameters. The values will change as the estimation progresses, especially with the estimation of the summed cross-correlated terms. The resulting yaw drag terms (Table 14) were used in the initial parameter population for the GA optimizer.

Table 14: The yaw-drag results used for initial estimation.

Yaw Parameters	
N_r	1745.1
N_{rr}	17.6

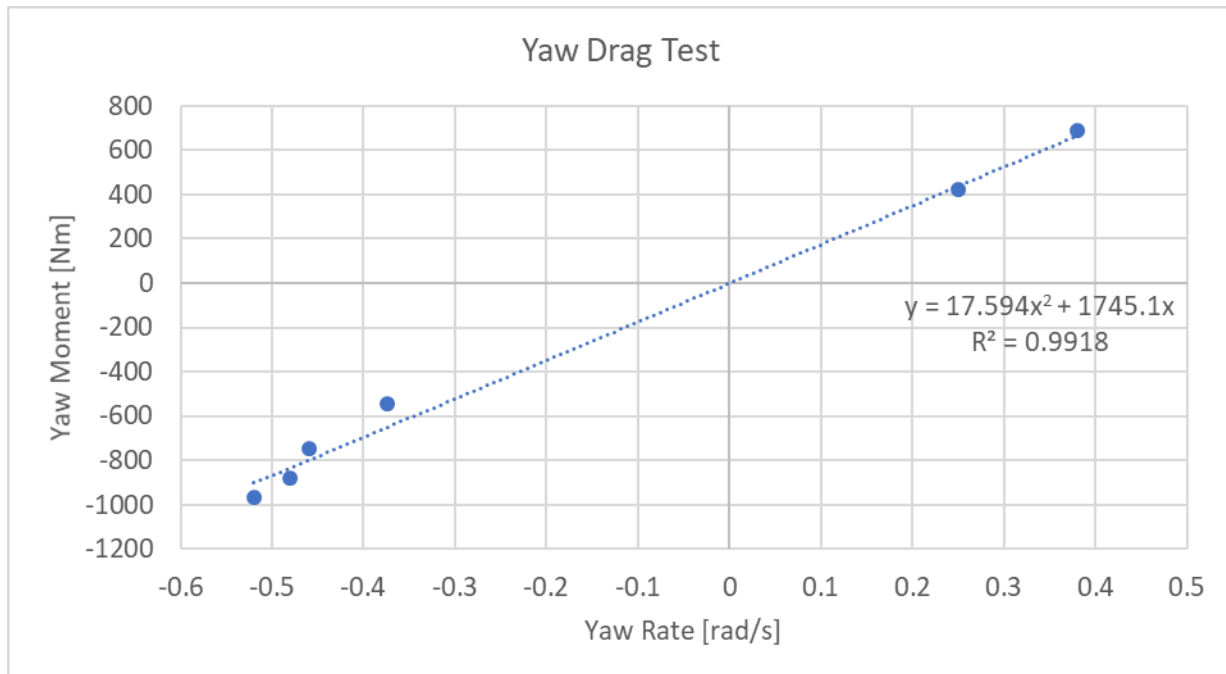


Figure 22: Yaw drag curve-fit from the steady-state test ranging from -0.55 to 0.38 rad/s.

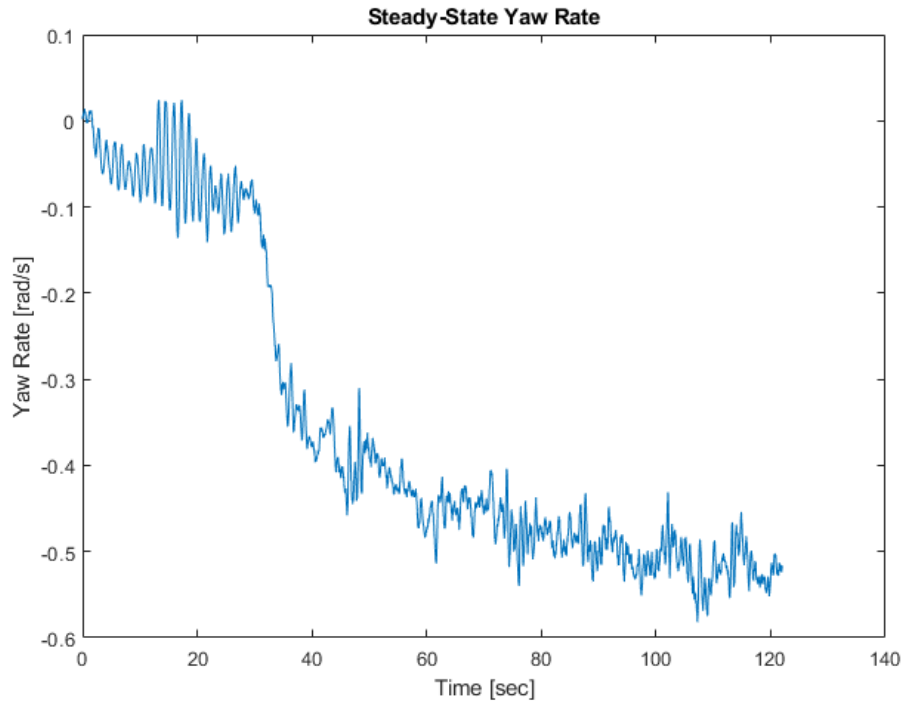


Figure 23: An example of a spin test yaw-rate states (Appendix 1.2.1)

4.4. Genetic Algorithm Fit (No Surge-Yaw Coupling)

The results from the steady-state tests are used to initialize the estimation for the sway-yaw coupled model (Table 15). The following estimations will follow three serialized fits using the same set of randomly assorted training data sets per optimization. The maneuvering training data sets used are the stopping test (Appendix 1.3.10), turning test (Appendix 1.4.1), and zig-zag test (Appendix 1.5.1). The results of the optimizations are displayed in state graphs that compares the data set (blue line) to the two simulations that have disturbances (black dashed line) and ignore disturbances (orange line).

Table 15: Initial population to use for GA fitting.

Wind		Surge		Sway		Yaw	
c_x	0	$X_{\dot{u}}$	0	$Y_{\dot{u}}$	0	$N_{\dot{u}}$	0
c_y	0	X_u	490.5	Y_u	0	N_u	0
c_z	0	X_{uu}	-25.88	Y_{uu}	0	N_{uu}	0
		$X_{\dot{v}}$	0	$Y_{\dot{v}}$	0	$N_{\dot{v}}$	0
		X_v	0	Y_v	0	N_v	0
		X_{vv}	0	Y_{vv}	0	N_{vv}	0
				Y_{vr}	0	N_{vr}	0
		$X_{\dot{r}}$	0	$Y_{\dot{r}}$	0	$N_{\dot{r}}$	0
		X_r	0	Y_r	0	N_r	1745.1
		X_{rr}	0	Y_{rr}	0	N_{rr}	17.6
				Y_{rv}	0	N_{rv}	0

4.4.1. GA “No Surge-Yaw” Fit #1

The first fit showed two notable details: the surge fitting and the offsets in sway and yaw states. The optimizer generated the parameters defined in Table 16 after the initial parameter set.

Table 16: Parameters estimated from the Lake Ashby set of tests for Fit #1.

Wind		Surge		Sway		Yaw	
c_x	0.05	$X_{\dot{u}}$	516	$Y_{\dot{u}}$	0	$N_{\dot{u}}$	0
c_y	9E-4	X_u	120	Y_u	0	N_u	0
c_z	-4E-4	X_{uu}	85	Y_{uu}	0	N_{uu}	0
		$X_{\dot{v}}$	0	$Y_{\dot{v}}$	613	$N_{\dot{v}}$	-2617
		X_v	0	Y_v	550	N_v	1499
		X_{vv}	0	Y_{vv}	1713	N_{vv}	-3685
				Y_{vr}	-2734	N_{vr}	2663
		$X_{\dot{r}}$	0	$Y_{\dot{r}}$	-1768	$N_{\dot{r}}$	4423
		X_r	0	Y_r	2761	N_r	5893
		X_{rr}	0	Y_{rr}	1007	N_{rr}	-3262
				Y_{rv}	111.6	N_{rv}	-670

The surge fit showed a visible fit to the stopping test with an average surge error over the mentioned training data sets. The average surge error is 0.154 m/s over the data sets mentioned above. The error in Figure 24 shows less estimated drag resistance of the model compared to the actual WAMV. The simulated wind disturbance starts to show promising results by following the

dips in surge speed as the WAMV rotates, even if the estimation is out of phase. The same parameters on the stopping test (Figure 25), the surge fits the steady-state speeds in both simulation sets, with the wind underestimating the drift speed.

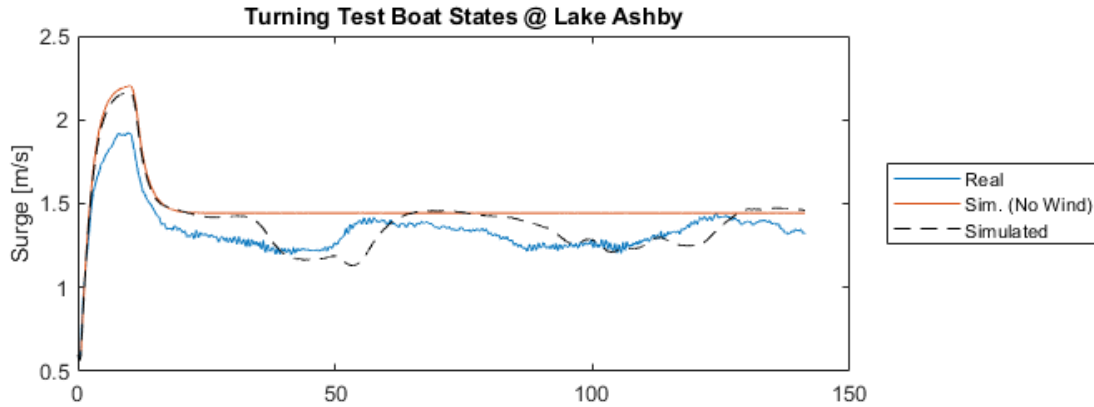


Figure 24: Surge result from a turning test validation set for Fit #1.

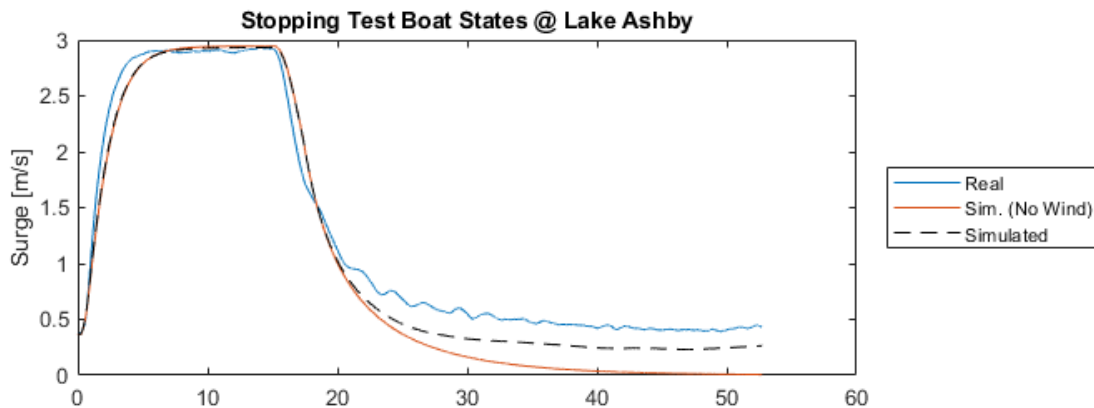


Figure 25: Surge result from a stopping test validation set for Fit #1.

The model on the sway and yaw states show different results coupling results. The yaw rate moves in the same direction that the turning test provides. The wind moment does not change the rate much since the disturbance simulation follows the non-disturbance simulation, but it does show a bump around the 50 sec mark. The sway velocity is showing a constant offset with a constant moment and with almost no response due to wind disturbance. Given the same disturbance bump around the 50 sec mark, the wind force pushes with the sway velocity in the

same direction by about 15 N. Excluding the modelled disturbances, the drag due to sway effects is too small to allow a significant error of approximately 0.2 rad/s.

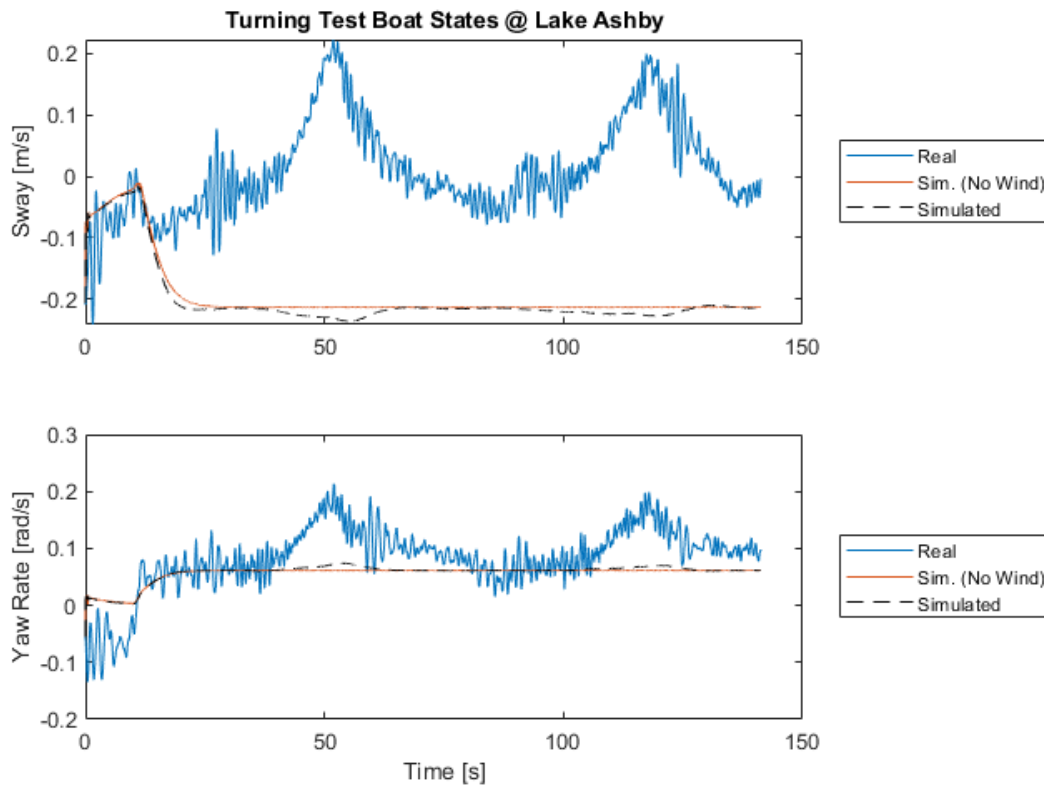


Figure 26: Sway and Yaw Rate result from a turning test validation set for Fit #1.

The zig-zag test (Figure 27) displays the effects of the dynamic yaw rate effects. The sway state allows large changes in sway over low yaw rates that do not have time to steady out. The yaw states estimated too much required drag given the input moments. The parameters need to move towards more sway-drag and less yaw-drag. The average sway and yaw rate errors over the validation sets were 0.12 m/s and 0.08 rad/s.

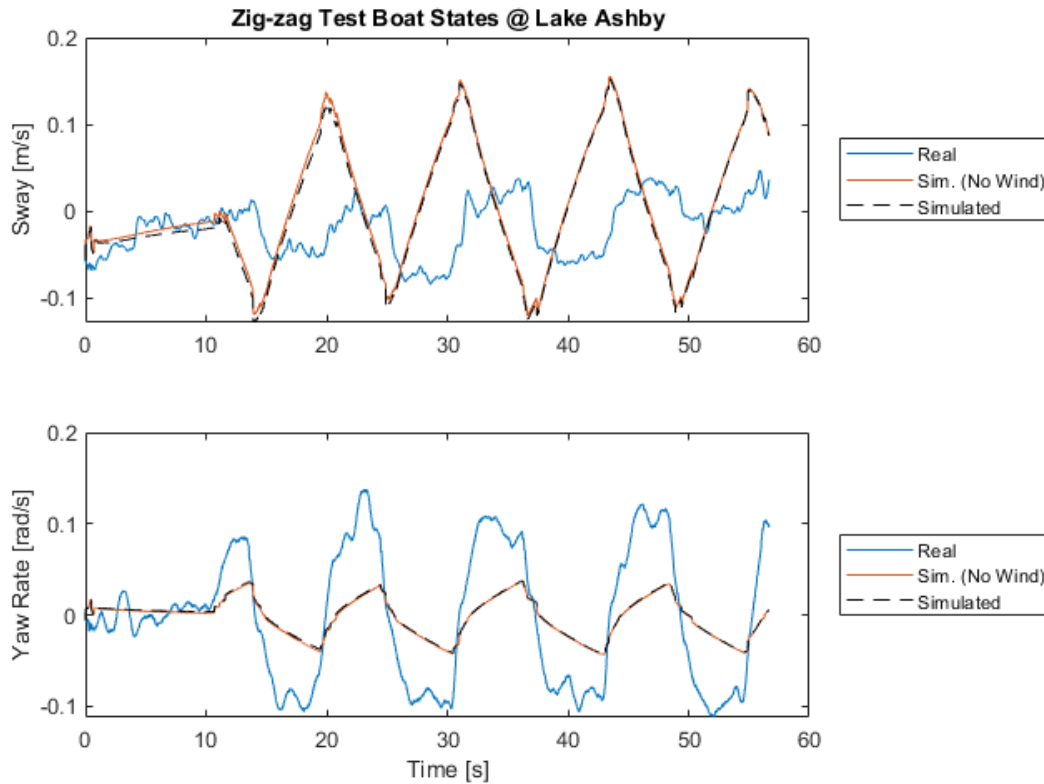


Figure 27: Sway and Yaw Rate result from a zig-zag test validation set for Fit #1.

With consideration to the yaw-sway states, the yaw-drag limits in the optimizer need to be increased to allow a better fit to the average states in the data set. Although the yaw states show heavy drag, that could be partially due to the sway coupling effects as well. Expanding the drag coupling and the added mass terms' limits can allow the optimizer to change those parameters more freely. The surge state bounds were expanded a little to allow for some improvement if necessary but appeared to fit well currently. The magnitudes for the wind coefficients are expanded to allow more potential impact that may occur due to the wind in the data sets as well. The bounds to be expanded are shown in

Table 17.

Table 17: Parameter limits used for the optimization for Fit #2.

Wind		Surge		Sway		Yaw	
c_x	{0.01, 0.15}	$X_{\dot{u}}$	{500, 600}	$Y_{\dot{u}}$	0	$N_{\dot{u}}$	0
c_y	{9E-4, 0.01}	X_u	{0, 500}	Y_u	0	N_u	0

c_z	{-0.01 0.01}	X_{uu}	{-100, 100}	Y_{uu}	0	N_{uu}	0
		$X_{\dot{v}}$	0	$Y_{\dot{v}}$	{-2500, 2500}	$N_{\dot{v}}$	{-4500, 4500}
		X_v	0	Y_v	{0, 4500}	N_v	{0, 4500}
		X_{vv}	0	Y_{vv}	{-6000, 6000}	N_{vv}	{-6500, 6500}
				Y_{vr}	{-6000, 6000}	N_{vr}	{-6500, 6500}
		$X_{\dot{r}}$	0	$Y_{\dot{r}}$	{-3500, 3500}	$N_{\dot{r}}$	{-4500, 4500}
		X_r	0	Y_r	{-4500, 4500}	N_r	{0, 6500}
		X_{rr}	0	Y_{rr}	{-4500, 4500}	N_{rr}	{-6500, 6500}
				Y_{rv}	{-4500, 4500}	N_{rv}	{-6500, 6500}

4.4.2. GA “No Surge-Yaw” Fit #2

This optimizer run of Fit #2 has not changed the surge parameters but has changed the wind and sway-yaw parameters by significantly increasing the overall drag primarily from the following states: Y_v , Y_{vv} , N_r , and N_{rr} .

Table 18: Parameters estimated from the Lake Ashby set of tests for Fit #2.

Wind		Surge		Sway		Yaw	
c_x	0.05	$X_{\dot{u}}$	516	$Y_{\dot{u}}$	0	$N_{\dot{u}}$	0
c_y	0.001	X_u	120	Y_u	0	N_u	0
c_z	-0.001	X_{uu}	85	Y_{uu}	0	N_{uu}	0
		$X_{\dot{v}}$	0	$Y_{\dot{v}}$	-613	$N_{\dot{v}}$	617
		X_v	0	Y_v	3550	N_v	1473
		X_{vv}	0	Y_{vv}	5715	N_{vv}	3690
				Y_{vr}	2732	N_{vr}	-3666
		$X_{\dot{r}}$	0	$Y_{\dot{r}}$	-2774	$N_{\dot{r}}$	424
		X_r	0	Y_r	-762	N_r	0
		X_{rr}	0	Y_{rr}	1005	N_{rr}	-6262
				Y_{rv}	1114	N_{rv}	-2663

The sway and yaw rate states are converging closer to the data sets. The average sway and yaw error between the data sets are 0.057 m/s and 0.025 rad/s for the turning data set. The sway state captures the average drag without the effects rotation and wind acting as a major effect. The simulated model matches closer to the mean motion of the yaw rate, including disturbances effects, as shown in the turning test (Figure 28). The drag in sway is too high to see any difference in the sway at steady-state yaw rates.

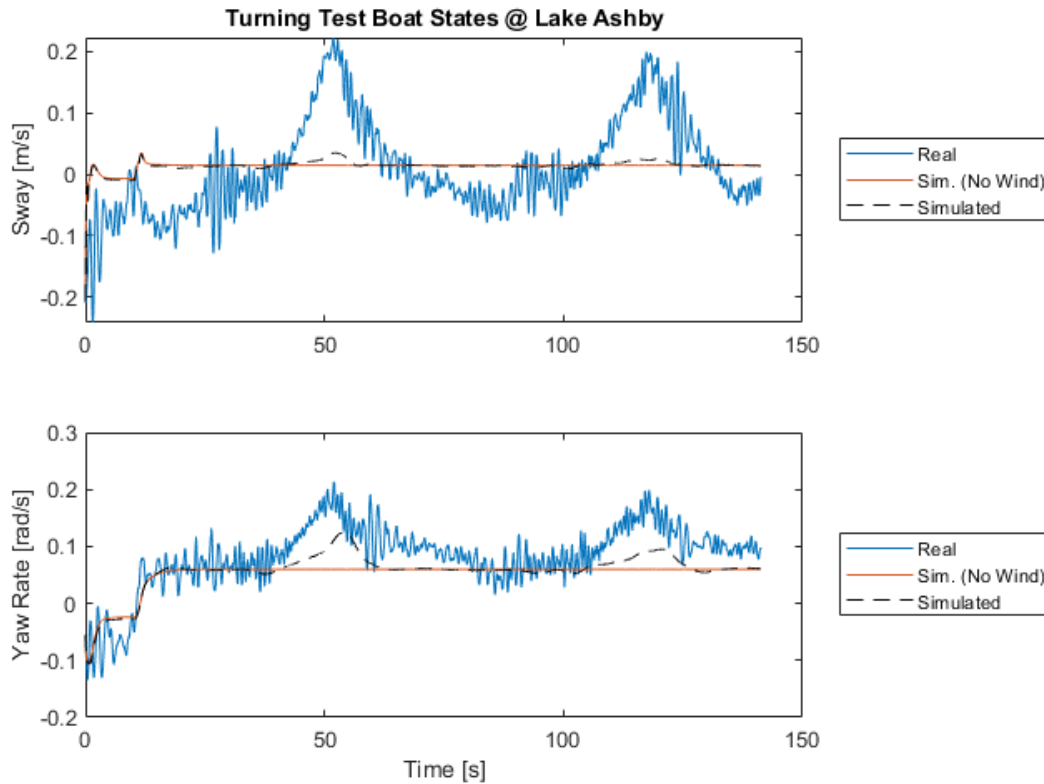


Figure 28: Sway and Yaw Rate result from a turning test validation set for Fit #2.

Looking at the zig-zag test, the sway is shown to fit closer to dynamic changes in yaw rates. Figure 29 shows the sway and yaw rate states for the zig-zag test where the simulated sway follows the direction of the data set. The initial jump of the expected sway speed is from the dynamic rate change to the steady-state drag before the input switches directions. This is from the amount of sway and yaw drag due to its primary states such as Y_v , Y_{vv} , N_r , and N_{rr} . The primary drag parameters limit the total amount of observed motion that the WAMV has. The spin test has shown a potential 0.5 rad/s due to the propulsion system; this estimation of the model maxes out around 0.15 rad/s at the same inputs. The primary drag parameters need to be reduced to handle the loss in motion, and the rest of the terms, coupled and added mass, need to be adjusted by increasing the limits of the coupled states.

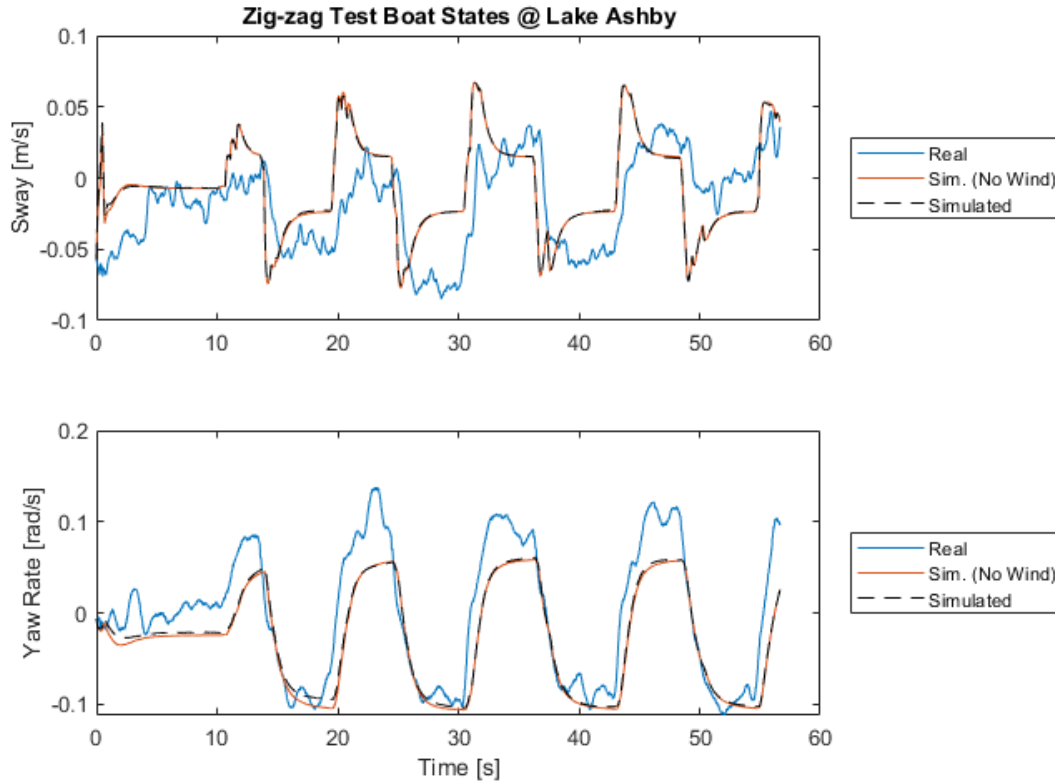


Figure 29: Sway and Yaw Rate result from a zig-zag test validation set for Fit #2.

Table 19: Parameter limits used for the optimization for Fit #3.

Wind	Surge	Sway	Yaw				
c_x	{0.025, 0.075}	$X_{\dot{u}}$	{516}	$Y_{\dot{u}}$	0	$N_{\dot{u}}$	0
c_y	{9E-4, 0.001}	X_u	{120}	Y_u	0	N_u	0
c_z	{-0.01, 0}	X_{uu}	{85}	Y_{uu}	0	N_{uu}	0
	$X_{\dot{v}}$	0	$Y_{\dot{v}}$	{-2500, 2500}	$N_{\dot{v}}$	{-4500, 4500}	
	X_v	0	Y_v	{0, 4500}	N_v	{0, 4500}	
	X_{vv}	0	Y_{vv}	{-6000, 6000}	N_{vv}	{-5500, 5500}	
			Y_{vr}	{-9000, 9000}	N_{vr}	{-5500, 5500}	
	$X_{\dot{r}}$	0	$Y_{\dot{r}}$	{-3500, 3500}	$N_{\dot{r}}$	{-4500, 4500}	
	X_r	0	Y_r	{-4500, 4500}	N_r	{0, 6500}	
	X_{rr}	0	Y_{rr}	{-4500, 4500}	N_{rr}	{-5500, 5500}	
			Y_{rv}	{-9000, 9000}	N_{rv}	{-9000, 9000}	

4.4.3. GA “No Surge-Yaw” Fit #3

This model fit lowered its primary drag magnitudes with a greater focus on the coupling terms: Y_{vr} , Y_{rv} , N_{vr} , and N_{rv} (Table 20). The coupled yaw drag for sway is shown to have a large magnitude. The previous models have given noticeably stable results in surge and yaw rate with offsets in certain states. The parameters were reduced to handle the individual primary drag terms (i.e. yaw moment with relation to yaw rate).

Table 20: Parameters estimated from the Lake Ashby set of tests for Fit #3.

Wind		Surge		Sway		Yaw	
c_x	0.05	$X_{\dot{u}}$	516	$Y_{\dot{u}}$	0	$N_{\dot{u}}$	0
c_y	0.01	X_u	120	Y_u	0	N_u	0
c_z	-0.001	X_{uu}	85	Y_{uu}	0	N_{uu}	0
		$X_{\dot{v}}$	0	$Y_{\dot{v}}$	991	$N_{\dot{v}}$	288
		X_v	0	Y_v	884	N_v	552
		X_{vv}	0	Y_{vv}	696	N_{vv}	-1516
				Y_{vr}	8361	N_{vr}	-425
		$X_{\dot{r}}$	0	$Y_{\dot{r}}$	-1736	$N_{\dot{r}}$	4486
		X_r	0	Y_r	-202	N_r	1258
		X_{rr}	0	Y_{rr}	814	N_{rr}	-2884
				Y_{rv}	307	N_{rv}	-202

The turning test (Figure 30) shows the disturbance simulation following the peaks of the validation set in both the sway and yaw rate. The average error over the data set is 0.05 m/s and 0.028 rad/s with wind and the undisturbed simulation remaining at a visibly appropriate state throughout. The average sway velocity at steady-state appears to be around -0.05 m/s for a clockwise rotation, leaving the sway velocity estimating the wrong direction, only off by 0.065 m/s over the steady-state maneuver.

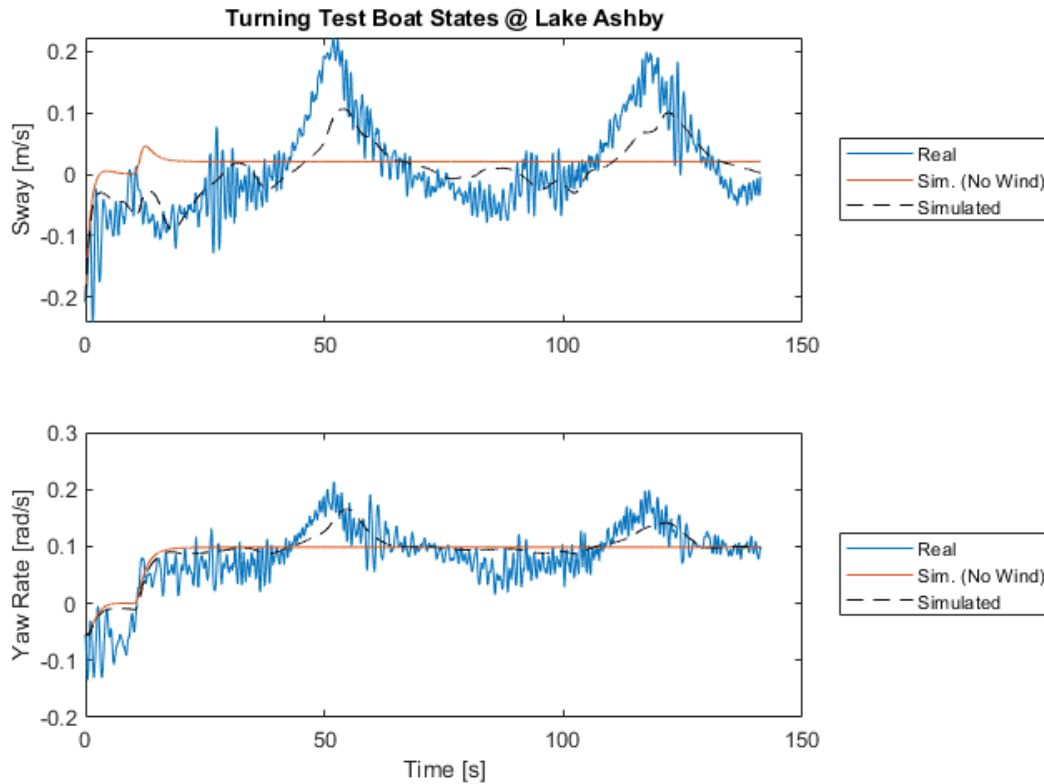


Figure 30: Sway and Yaw Rate result from a turning test validation set for Fit #3.

The zig-zag test (Figure 31) follows the same trend during dynamic change by follow the motion of the data sets. The yaw rate is following the peaks very well with the 2nd-order modeling of drag. The sway speed lessened the ramp of the curves that Fit #2 contained and is taking more shape. The sway drag may still be too heavy since after each peak, the drop-off curve appears to underestimate the magnitude of the data set. If the turn held longer like the turning test, it would show a need to lessen the steady-state sway drag. The average errors of the sway and yaw rate were 0.041 m/s and 0.05 rad/s over the zig-ag data set.

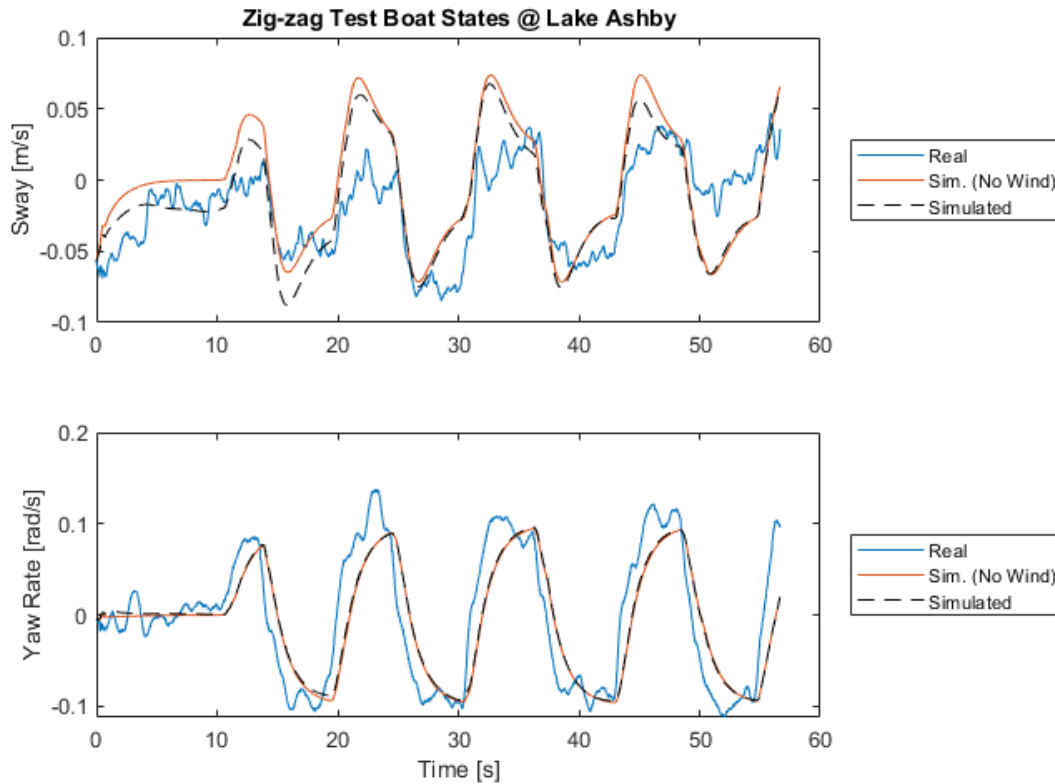


Figure 31: Sway and Yaw Rate result from a zig-zag test validation set for Fit #3.

4.4.4. GA “No Surge-Yaw” Validation Results

The previous parameters were checked for how well they fit with a set of validation results. The validation runs took a stopping, turning, zag-zag, and figure eight data set to verify how well this model handles its deviation between the available states. Table 21 shows the mean error over the entire data set including disturbances. The surge state did not change much since the initial value showed good responses with no change over the estimations. The sway and yaw rates made most of the improvements to capture the coupling that occurs. The errors made steady improvement with yaw rate making the tightest fits visibly to the data sets. The sway needed more estimation to capture its steady state and dynamic motion, but appear negligible in the validation data sets since the sway due to yaw rate hardly moved past 0.1-0.2 m/s. For use in simulation,

sway can be improved, but should operate fine in a differentially-driven configuration. Further optimization for this state would benefit station-keeping simulations.

Table 21: The mean error of the four different validation tests using the past three parameter sets.

Sets	Fit #1			Fit #2			Fit #3		
	X	Y	N	X	Y	N	X	Y	N
Stop	0.17	0.017	8.7E-3	0.17	0.01	0.021	0.17	0.011	8.5E-3
Turn	0.33	0.26	0.07	0.26	0.057	0.025	0.31	0.05	0.028
Zig-Zag	0.07	0.12	0.08	0.068	0.036	0.054	0.069	0.041	0.05
Figure-Eight	0.24	0.28	0.076	0.19	0.047	0.06	0.21	0.011	0.011

The figure-eight test was used for validation since it encompassed sections of all the tests and determined the disturbance drift over time. The figure-eight test shows steady improvement of data set error over the fits with the reference position and states shown in Figures 32 and 33. The position map is to show how close the simulation results performed visually with the drift due to wind showing up in the same direction as well. The simulation also drifted off from the real position due to round off errors and integration of small underestimation in the states.

With the wind disturbance, the inputs to the model were the actual relative wind velocities that the actual vessel experienced. These were used to help the optimizer's fit by adding what the vessel experienced rather than what the model would expect to experience. This means that based on the heading to the wind, the angle of attack, can accrue error in the simulated model which leads to expected offsets in states, but similar patterns. In the case of surge, Figure 33 shows approximated offset motion around 120 to 140 seconds where the states appear out of phase along the y-axis of the graph. And the physical state offset is due to the wind inputs during the turns being out of sync since the headings were diverging like the latter half of the positions in Figure 32.

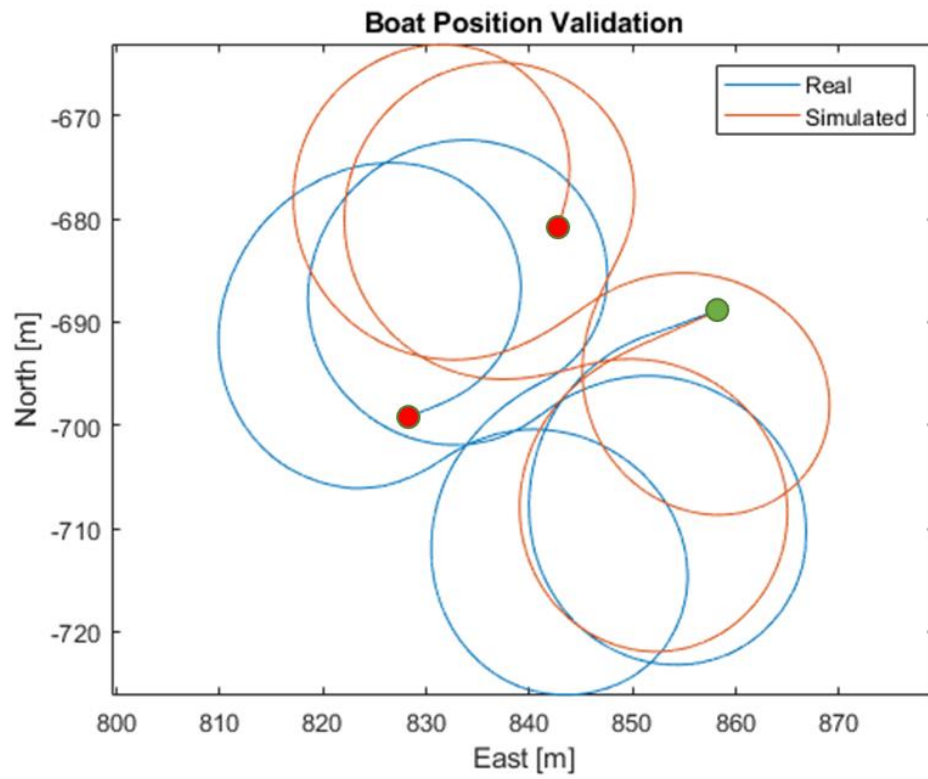


Figure 32: A comparison of Fit #3 of the position validation figure-eight test. The green circle show initial position and the red circles show its ending positions for the real and simulated positions.

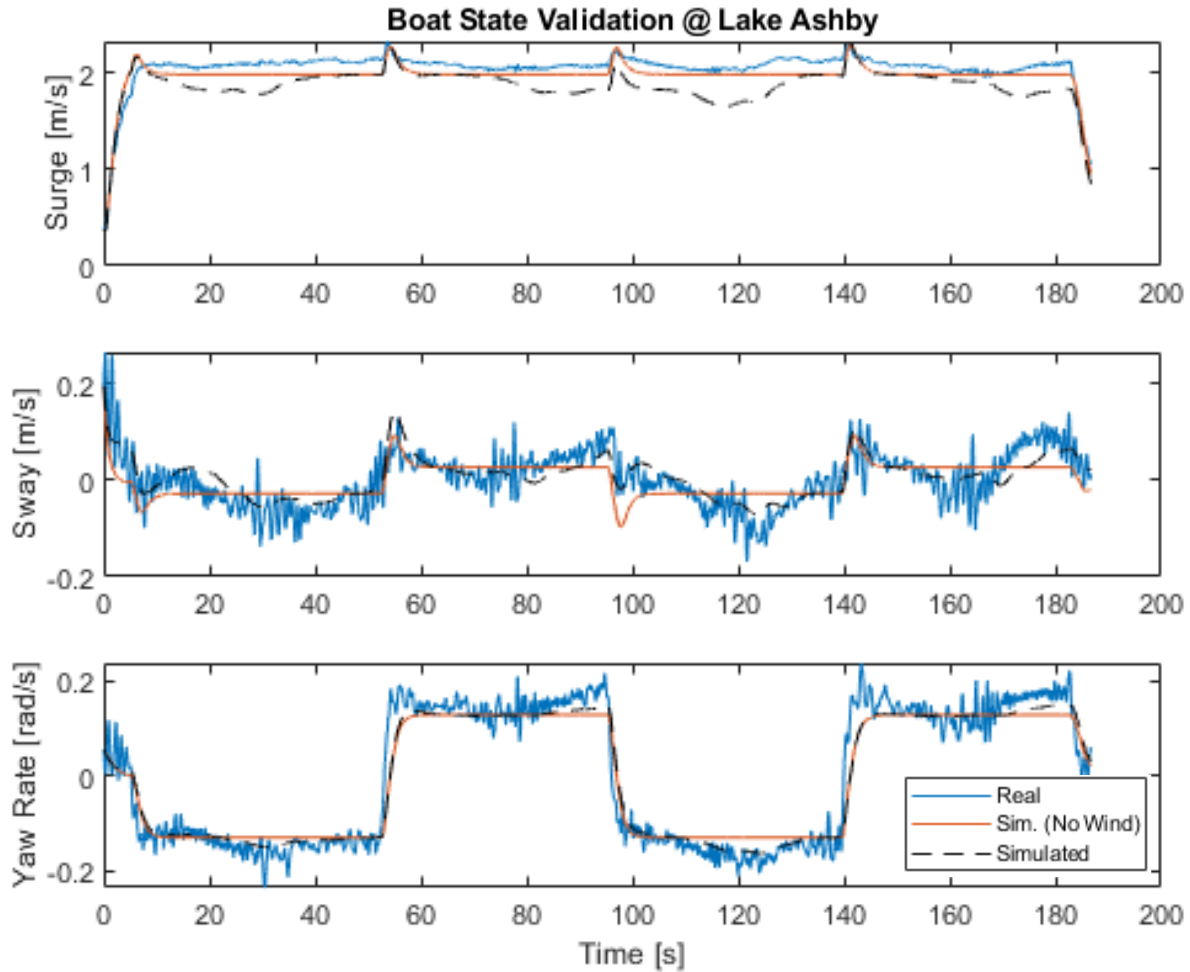


Figure 33: The figure-eight validation set using Fit #3's parameters.

4.5. Genetic Algorithm Fit (Full Coupling)

4.5.1. GA “Full” Fit

The full coupling takes the Fit #1 set of “No Surge-Yaw” coupling parameters as its initial parameters. This model has an addition of 12 extra parameters that make the manual supervision more difficult due to even more coupled effects. Table 22 shows the parameters estimated for the full coupling case.

Table 22: Parameter set for the full coupling set model.

Wind		Surge		Sway		Yaw	
c_x	0.05	$X_{\dot{u}}$	592.6	$Y_{\dot{u}}$	0	$N_{\dot{u}}$	146
c_y	0.001	X_u	39	Y_u	0	N_u	238.5
c_z	-0.001	X_{uu}	112.6	Y_{uu}	0	N_{uu}	-134.5
		$X_{\dot{v}}$	201	$Y_{\dot{v}}$	1164	$N_{\dot{v}}$	1810.5
		X_v	100	Y_v	579.5	N_v	3360
		X_{vv}	-72.4	Y_{vv}	1466.5	N_{vv}	-1926.1
				Y_{vr}	-175	N_{vr}	-2292
		$X_{\dot{r}}$	195	$Y_{\dot{r}}$	1892	$N_{\dot{r}}$	1349
		X_r	101	Y_r	2665	N_r	44.5
		X_{rr}	-89	Y_{rr}	2068	N_{rr}	1277
				Y_{rv}	-1706	N_{rv}	-455.4

Compared to the “No Surge-Yaw” coupling, the surge still represents similar motion, but with the surge affecting the other states. This test considered the effects of the sway forces on the surge velocity to be negligible due to it being not directly driven. The surge during the figure-eight test (Figure 34) showed consistency with minimal changes in speed (~ 0.3 m/s) for mean simulated surge and larger dips due to wind.

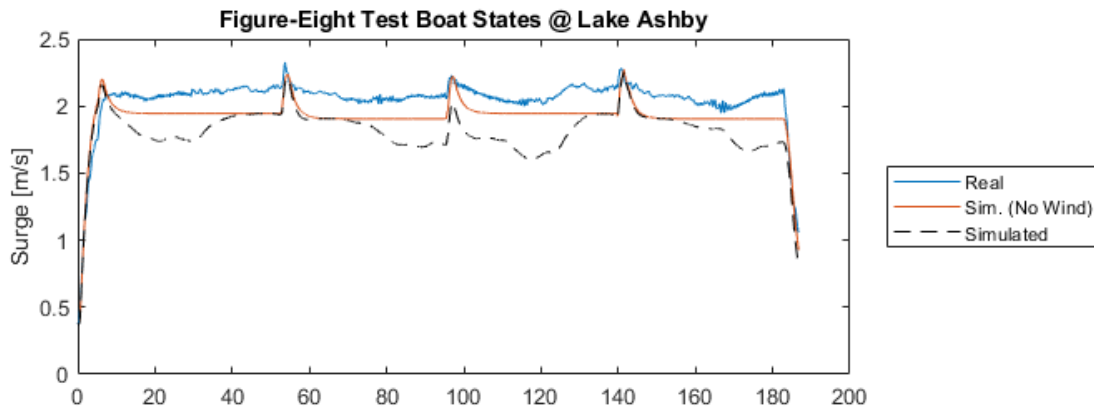


Figure 34: Surge result from a figure-eight test validation set for the Full fit.

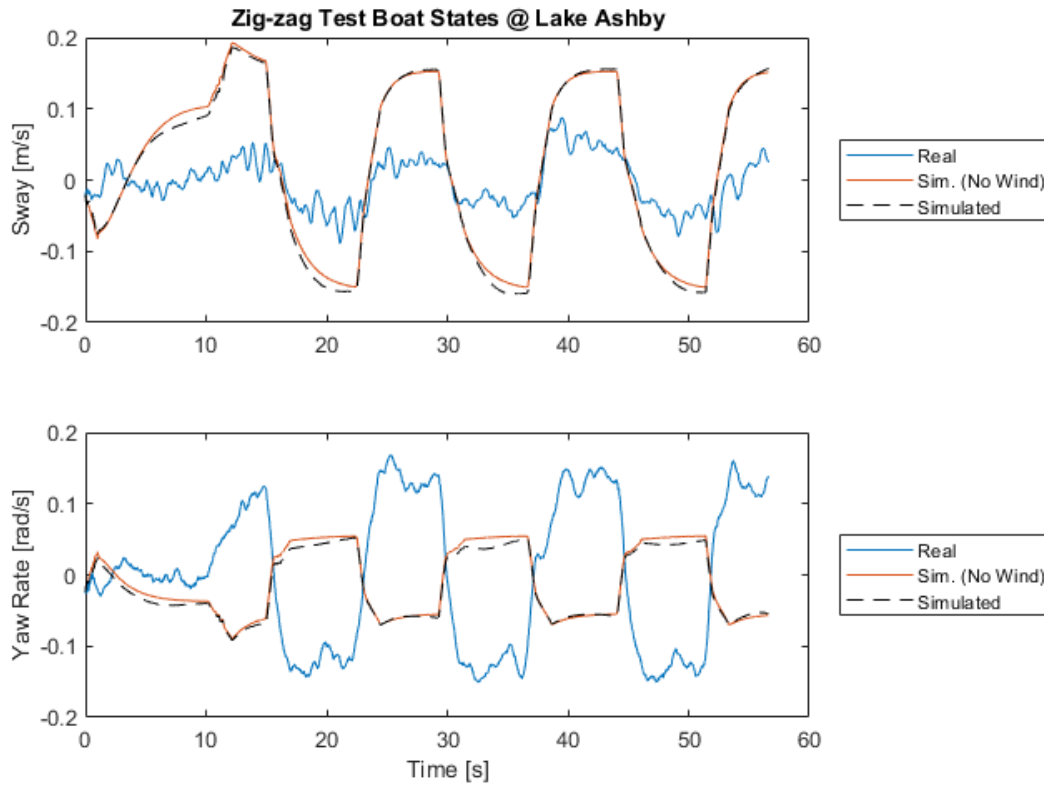


Figure 35: Sway and Yaw Rate result from a zig-zag test validation set for the Full fit.

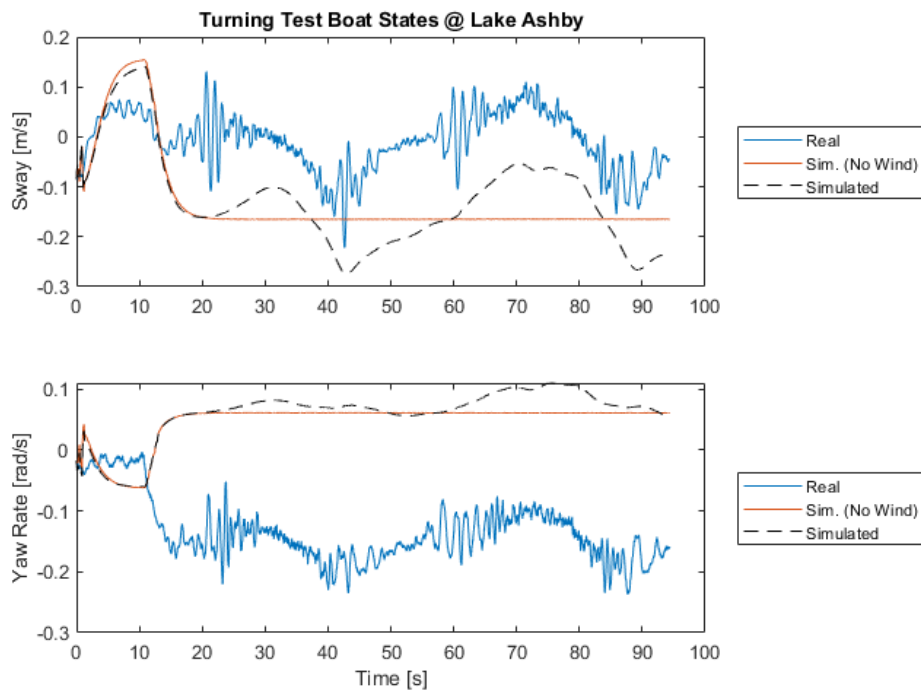


Figure 36: Sway and Yaw Rate result from a turning test validation set for the Full fit.

The two glaring data sets (Figures 35 and 36) that show issues in estimating the “Full” model is the lack of sway drag and the reversed direction of the yaw rate. In both the turning and zig-zag data sets the sway overshoots the state, but the wind compensates for the correct disturbance effects during steady state rotations. Any other jumps in dynamic effects are difficult to quantify due to the yaw rate opposing the data set. The yaw rate state shows both too much drag moment to approximately 40% of the magnitude of the data set. While both yaw rate problems can be fixed, the parameter(s) that need to be changed and the method to detect them are not straightforward. In the case of the drag magnitude, the added terms from the surge coupling means that more values depend on a single state changing and picking them out is difficult. Running the optimizer more should find a way to converge onto the solution, but the next problem is the opposing direction of the yaw rate state. The optimizer calculates a single, positive number that determines how the optimizer can randomly evolve per generation. This error is a sum squared error of all the weighted states. Which means the error magnitude may be larger, but it may not track down the parameter(s) that can be altered to fix the issue.

4.5.2. GA “Full” Validation Results

The coupling was run through the same validation sets as the “No Surge-Yaw” model to determine the average error. Due to the poor performance, the error is substantially higher in the yaw rate cases with sway not looking bad compared to the “No Surge-Yaw” fit. As stated earlier, the model will need more time to optimize and stochastically leave this case or another method needs to be implemented to handle error sign issues.

Table 23: The mean error of the four different validation tests between the “No Surge-Yaw” and “Full” coupling cases.

Sets	Fit #3			Full		
	X	Y	N	X	Y	N
Stop	0.17	0.011	8.5E-3	0.162	0.053	0.021
Turn	0.31	0.05	0.028	0.368	0.138	0.211
Zig-Zag	0.069	0.041	0.05	0.074	0.093	0.149
Figure-Eight	0.21	0.011	0.011	0.268	0.134	0.208

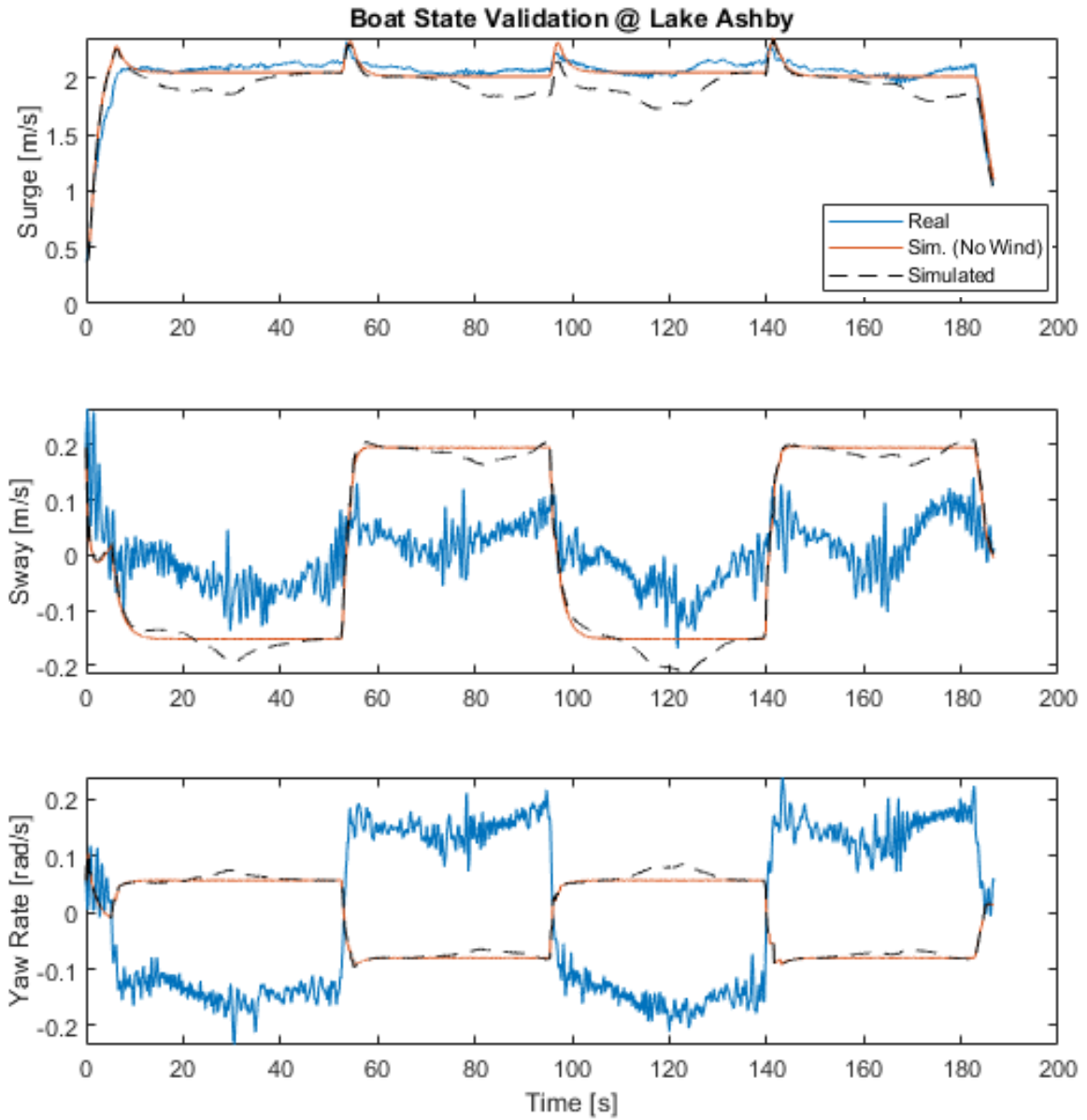


Figure 37: The full coupling simulation of the figure-eight validation test.

4.6. Comparison: “No Surge-Yaw” vs. “No Coupling”

The results in fit #3 for the “No Surge-Yaw” gave good results compared to the validation data sets. The estimation of the parameters was not intuitive for every parameter but was necessary to capture coupled-state effects of the sway and yaw states. Simplifying the model to a linearized form will make the model work in a basic sense, but with inaccurate coupling effects. The following show the linearized equations and the parameters used:

$$M_{AM} = \begin{bmatrix} X_{\dot{u}} & 0 & 0 \\ 0 & Y_{\dot{v}} & 0 \\ 0 & 0 & N_{\dot{r}} \end{bmatrix} \quad (73)$$

$$C_{AM}(\vec{v}) = \begin{bmatrix} 0 & 0 & Y_{\dot{v}}v + Y_{\dot{r}}r \\ 0 & 0 & -X_{\dot{u}}u \\ -Y_{\dot{v}}v - Y_{\dot{r}}r & X_{\dot{u}}u & 0 \end{bmatrix} \quad (74)$$

$$D_L = \begin{bmatrix} X_u & 0 & 0 \\ 0 & Y_v & 0 \\ 0 & 0 & N_r \end{bmatrix} \quad (75)$$

$$D_{NL}(\vec{v}) = \begin{bmatrix} X_{uu}|u| & 0 & 0 \\ 0 & Y_{vv}|v| & 0 \\ 0 & 0 & N_{rr}|r| \end{bmatrix} \quad (76)$$

Table 24: Best set of parameters that fit the training and validation set of data.

Wind		Surge		Sway		Yaw	
c_x	0.05	$X_{\dot{u}}$	516	$Y_{\dot{u}}$	0	$N_{\dot{u}}$	0
c_y	0.01	X_u	120	Y_u	0	N_u	0
c_z	-0.001	X_{uu}	85	Y_{uu}	0	N_{uu}	0
		$X_{\dot{v}}$	0	$Y_{\dot{v}}$	991	$N_{\dot{v}}$	0
		X_v	0	Y_v	884	N_v	0
		X_{vv}	0	Y_{vv}	696	N_{vv}	0
				Y_{vr}	0	N_{vr}	0
		$X_{\dot{r}}$	0	$Y_{\dot{r}}$	0	$N_{\dot{r}}$	4486
		X_r	0	Y_r	0	N_r	1258
		X_{rr}	0	Y_{rr}	0	N_{rr}	-2884
				Y_{rv}	0	N_{rv}	0

The parameters are based on the “No Surge-Yaw” fit that relies on the cross-terms, such as Y_{rv} and N_{rv} , to control parts of the dynamic drag. The linearized model removes the cross-terms,

so the result will not work for every input force and initialized state in a data set. Figure 38 shows how the model responded to a zig-zag data set (Appendix 1.5.14). This data set was shown because turning and some zig-zag data sets failed to simulate due to the parameters in this model and the respective inputs applied.

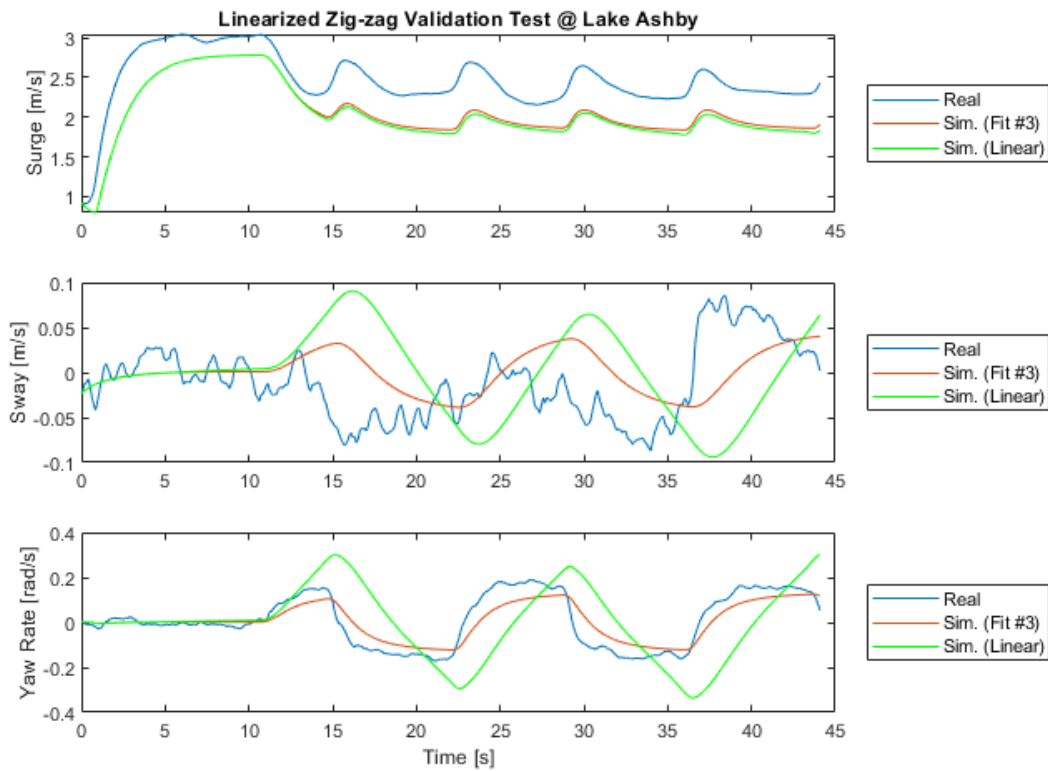


Figure 38: Results of a zig-zag data set using the linearized EOMs and compared to the “No Surge-Yaw” #3 fit.

The linearized model tries to fit the data primarily due to the second order drag with less effect available from Coriolis. The mean errors for the model are shown in Table 25. The surge simulation does not change much since the surge had minimal effects from Coriolis originally. The sway and yaw rate show low drag effects due to the linear simulation state moving linearly without signs of settling to a constant state before a change in input moment. The sway state is only affected by the surge speed in the Coriolis matrix. There is a phase delay shown due to the damping effects of the linearized added mass being too high as well. Overall, the drag could be estimated more to

estimate the linearized effects, but when the states are moving in tandem, all of the states can have higher speeds due to no coupling drags to fit the coupling effects that the “No Surge-Yaw” states could find. This method is useful for basic motions of vessels with a need for simple motion without accurate results in complex maneuvers.

Table 25: Mean error linearized coupling results.

Linearized Model			
Error	X	Y	N
Zig-Zag	0.52	0.06	0.17

4.7. Down-Sampling Data Sets for Future Estimation

The results shown in the previous were done with the localization sampling rate of 100 ± 2 Hz. The amount of time was substantial for the total number of samples per data set to simulate, especially with the larger logs like the figure-eight tests. Looking back to the dynamics presented in the data sets, the motion of the WAMV does not change quickly over time with smooth transitions that follow quadratic curves mostly. The logs were later down-sampled to determine if any information is lost in capturing these motion curves. The sample was taken from specified time steps and not filtered beforehand; the filtering is performed on the Pinpoint system natively.

The two examples are a 10 Hz and 1 Hz down-sampling of a zig-zag test (Appendix 1.5.14: Zig-Zag Set 14). The 10 Hz (Figure 39) sampling was chosen to see if there would be a noticeable difference by moving down by an order of magnitude. The data looks similar overall with small deviations in sway and yaw rate as shown in Figure 41. The 1 Hz (Figure 40) sampling was taken down by another order of magnitude to see any change. The noise is more visible in sway and yaw rate with the yaw rate missing some key dynamic points. The curves around the 15 and 16 second mark are not estimated at the 1 second time step, so the down sampling has been taken too far and is showing aliasing effects.

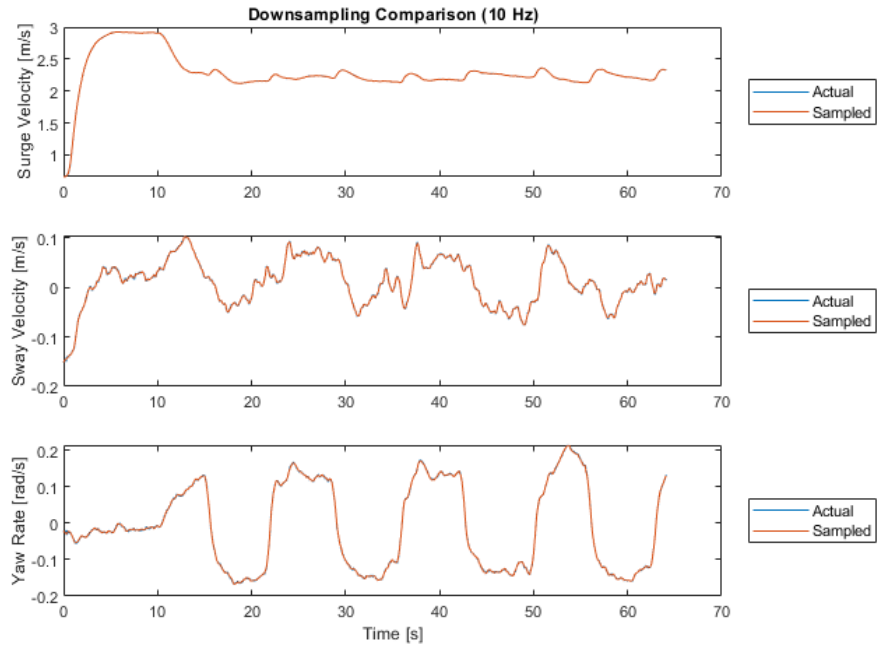


Figure 39: Comparison of a zig-zag test of the 100Hz vs. the 10 Hz data set.

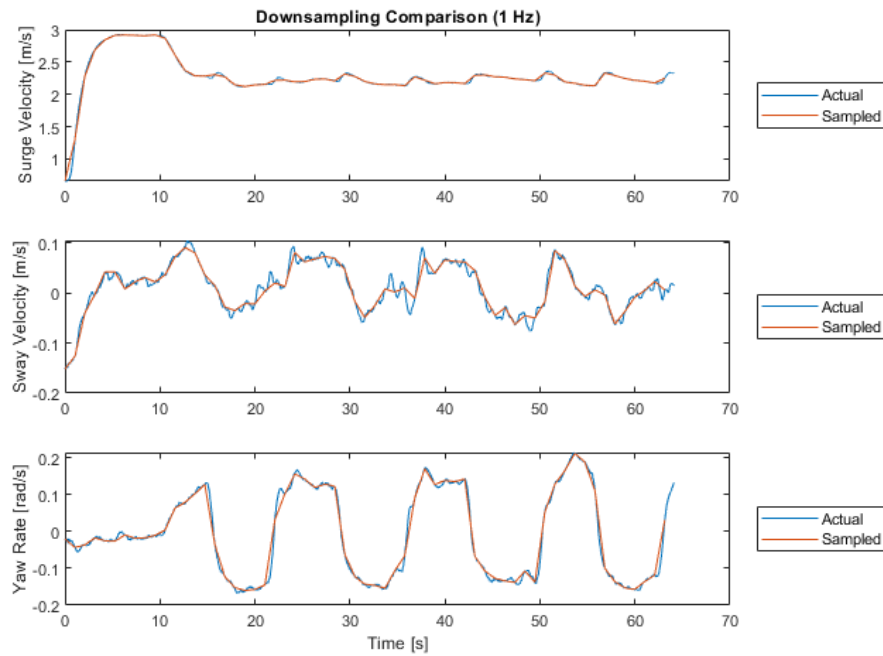


Figure 40: Comparison of a zig-zag test of the 100Hz vs. the 1 Hz data set.

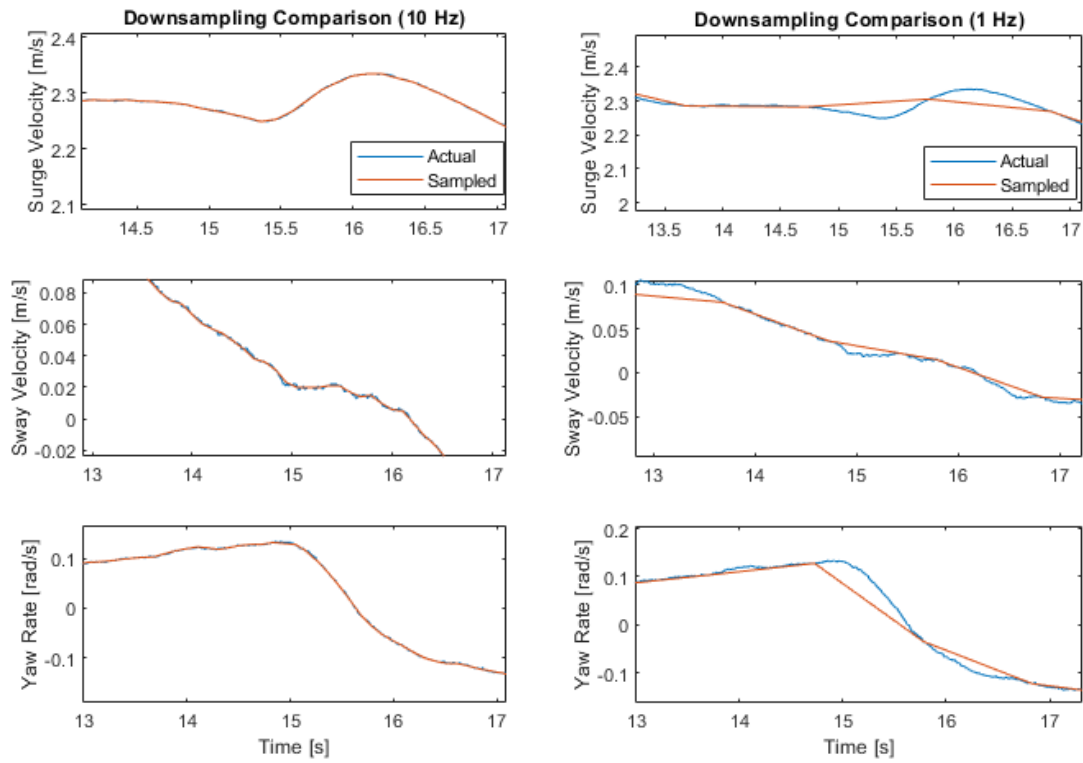


Figure 41: Zoomed in differences in the 100Hz data set with 10 Hz and 1 Hz.

Chapter V

Conclusions & Future Work

5.1. Conclusion

The methodology for estimating a marine vessel model produced parameters useful for future simulation work with careful supervision. The model that worked best was the “No Surge-Yaw” model that required minimal effects of coupling-states on surge speed with the sway speed and yaw rate requiring coupling to handle steady-state and dynamically changing turns.

The optimizer worked for searching the space for the full sets of parameters. As each fit was completed, the optimizer would find better estimates of the parameters that would fit each data set. The fits were better because some parameters were not as explored or continually expanded during the optimization time. After each fit, manual tuning of parameters was performed to approximate expected motion and those values were run through the optimizer for the next fit. The reason for needing to tune the parameters stemmed from the single-valued error that was calculated to determine if the solution was converging to a minimum. The error did not consider each individual state, nor their signs, so the optimizer would need to run longer to eventually catch these errors randomly.

With the models used, the “No Surge-Yaw” case was better than both the “Full” and “No Coupling” models. The “No Surge-Yaw” model had fewer nonlinearities than the “Full” model to consider for estimation and was easier to determine the physical insight of where some drag and damping effects were occurring. The “Full” model did not benefit from this optimization due to estimations reversing the states after an estimation fit. The parameters were not as intuitive to change with 12 extra parameters included. This model requires another methodology to capture that error in a multi-objective format. The “No Coupling” model was tested based on how other papers release their coefficients for the 2nd-order Fossen model. This model is only used for basic

control law estimations and basic movements for a marine vessel but does not capture any expected coupling effects.

The results of the “No Surge-Yaw” model did converge or steady out per optimization. The differences from Fit #1 to Fit #2 showed an overall convergence with all states. The fits lowered errors with the steady-state and dynamic sway velocities and yaw rates. The surge results remained steady with small improvements. The differences from Fit #2 to Fit #3 made some improvements to sway and yaw-rate, but error steadied out overall. The validation from the figure-eight revealed a good fit that nearly overlapped with that data set with small underestimations in the dynamic states.

Overall, the optimizer used did generate a set of parameters for a working system model. The methodology determined the parameters for a coupled-state dynamic model from nearly no initialization besides four initial values to help it along. The parameters from Fit #3 are the results that are being used for the WAMV16’s final model in this thesis work.

5.2. Future Work

The methodology helped move the estimation of the vessel along, but the reversed error was not captured appropriately. A method to be looked at in the future is a form of multi-objective optimization. Quantifying the different state errors as a vector may be beneficial to searching for a minimization solution to estimate or find intermittent issues in parameter between optimization runs.

Given a previously existing model, another set of optimizers can be more effective to improve a model. A nonlinear, grey-box modelling method can be used to search the space as a local optimizer with to find a local parameter solution. There is a work that Mask, in [3], developed a grey-box estimator in MATLAB using Fossen’s 3rd-order, truncated Taylor series model using a nonlinear grey-box method.

The input model was simplified due to a lack of actual equipment to estimate the thruster model. The thruster model needs to consider the torque and inertia of the propeller, and the water flow effects that affect the thrust in general. One method to use for the future is a four-quadrant motor model [12], where the thrust and torque are estimated based on the water flow speed and the torque acting on the propeller. This model handles cases of station-keeping and “braking” effects like attempting to reverse while moving forward. This model is characterized based on different inlet water speeds, up to the maximum speed that is needed to estimate, and the speed of the propeller.

Bibliography

- [1] Marine Advanced Robotics, Inc., "WAMV-16 ASV," Marine Advanced Robotics, Inc., [Online]. Available: <http://www.wam-v.com/wam-v-16-asv>. [Accessed 2019 17 January].
- [2] T. I. Fossen, Handbook of Marine Craft Hydrodynamics and Motion Control, John Wiley & Sons Ltd., 2011.
- [3] J. L. Mask, "System Identification Methodology for a Wave Adaptive Modular Unmanned Surface Vehicle," Florida Atlantic University, Boca Raton, 2011.
- [4] E. I. Sarda, H. Qu, I. R. Bertaska and K. D. von Ellenrieder, "Station-keeping control of an unmanned surface vehicle exposed to current and wind disturbances," *Ocean Engineering*, vol. 127, pp. 305-324, 2016.
- [5] T. I. F. Tristan Perez, "Kinematic Models for Manoeuvring and Seakeeping of Marine Vessels," *Norwegian Society of Automatic Control*, pp. 19 - 30, 2007.
- [6] D. J. K. a. Y.-i. L. Jongho Shin, "Adaptive Path-Following Control for an Unmanned Surface Vessel Using an Identified Dynamic Model," *IEEE*, pp. 1143 - 1153, 2017.
- [7] W. B. Klinger, "Adaptive Controller Design for an Autonomous Twin-Hulled Surface Vessel with Uncertain Displacement and Drag," Florida Atlantic University, Boca Raton, 2014.
- [8] M. A. Abkowitz, Lectures on Ship Hydrodynamics – Steering and Maneuverability, Lyngby: Lyngby: Hydrodynamics Department, 1964.
- [9] K. K. Fedyaevsky and G. V. Sobolev, Control and Stability in Ship Design., Leningrad: State Union Shipbuilding Industry Publishing House, 1963.
- [10] H. Qu, "Wind Feed-forward Control of a USV," *IEEE*.

- [11] J. De Kat and J. W. Wichers, "Behavior of a Moored Ship in Unsteady Current, Wind and Waves," *Marine Technology*, vol. 28, pp. 251-264, 1991.
- [12] A. S. J. H. Andreas J. Häusler, "Four-Quadrant Propeller Modeling: A Low-Order Harmonic Approximation," in *IFAC Conference on Control Applications in Marine Systems*, Osaka, 2013.
- [13] E. I. Sarda, "Development of a USV Station-Keeping Controller," *IEEE*.
- [14] L. Ljung, "Perspectives on System Identification," vol. 34, no. 1, 2010.
- [15] K. Bendict and M. Kirchoff, "Explaining Ships Dynamic and Handling Using MATLAB & SIMULINK as Simulation Tool in Teaching Maritime Students," Warnemünde.
- [16] MathWorks, "MathWorks: What is a Global Optimization?," MathWorks, 2018. [Online]. Available: <https://www.mathworks.com/help/gads/what-is-global-optimization.html>. [Accessed 15 January 2019].
- [17] MathWorks, "What Is The Genetic Algorithm?," MathWorks, 2018. [Online]. Available: <https://www.mathworks.com/help/gads/what-is-the-genetic-algorithm.html>. [Accessed 15 January 2019].
- [18] J. Carr, "An Introduction to Genetic Algorithms," Whitman College, Walla Walla, 2014.
- [19] GeeksForGeeks, "Genetic Algorithms," GeeksForGeeks, [Online]. Available: <https://www.geeksforgeeks.org/genetic-algorithms/>. [Accessed 15 January 2019].
- [20] J. R. Banga and A. Gábor, "Robust and efficient parameter estimation in dynamic models of biological systems," *BMC Systems Biology*, 2015.
- [21] I. M. ORGANIZATION, "EXPLANATORY NOTES TO THE STANDARDS FOR SHIP MANOEUVRABILITY," INTERNATIONAL MARITIME ORGANIZATION, 2002.

- [22] American Bureau of Shipping, "Guide for Vessel Maneuverability," American Bureau of Shipping, Houston, 2006.
- [23] TORC Robotics, "PinPoint User Manual V1.4," TORC Robotics, Blacksburg, 2015.
- [24] WindSonic, "WindSonic User Manual," Gill Instruments, Lymington, 2019.
- [25] F. Jenne, "Simulation & Control Optimization of an autonomous sail boat," Swiss Federal Institute of Technology Zurich, Zürich, 2010.
- [26] T. I. Fossen, *Marine Control Systems: Guidance, Navigation, and Control of Ships, Rigs and Underwater Vehicles*, Marine Cybernetics, 2002.
- [27] M. Mousaviraad, "URANS Studies of WAM-V Multi-Body Dynamics in Calm Water and Waves," ResearchGate, Iowa City, 2013.
- [28] J. D. a. G. S. S. Arvind Pereira, "An Experimental Study of Station Keeping on an Underactuated ASV," *IEEE*, 22 February 2008.
- [29] L. W. J.-y. Z. Yu-lei Liao, "Backstepping dynamical sliding mode control method for the path following of the underactuated surface vessel," *ScienceDirect*, 2011.
- [30] H. W. T. T. J. R. Stefan Wirtensohn, "Parameter Identification and Validation Analysis for a Small USV," *IEEE*, 2015.
- [31] S. S. a. T. F. AJ. S~rensen, "DESIGN OF A DYNAMIC POSITIONING SYSTEM USING MODEL-BASED CONTROL," vol. 4, pp. 359-368, 1995.
- [32] C. R. Sonnenburg, "Modeling, Identification, and Control of an Unmanned Surface Vehicle," Virginia Polytechnic Institute and State University, 2012.
- [33] C. A. Poor, "SIMULINK modeling of a marine autopilot for TSSE ship designs," NAVAL POSTGRADUATE SCHOOL, 1996.

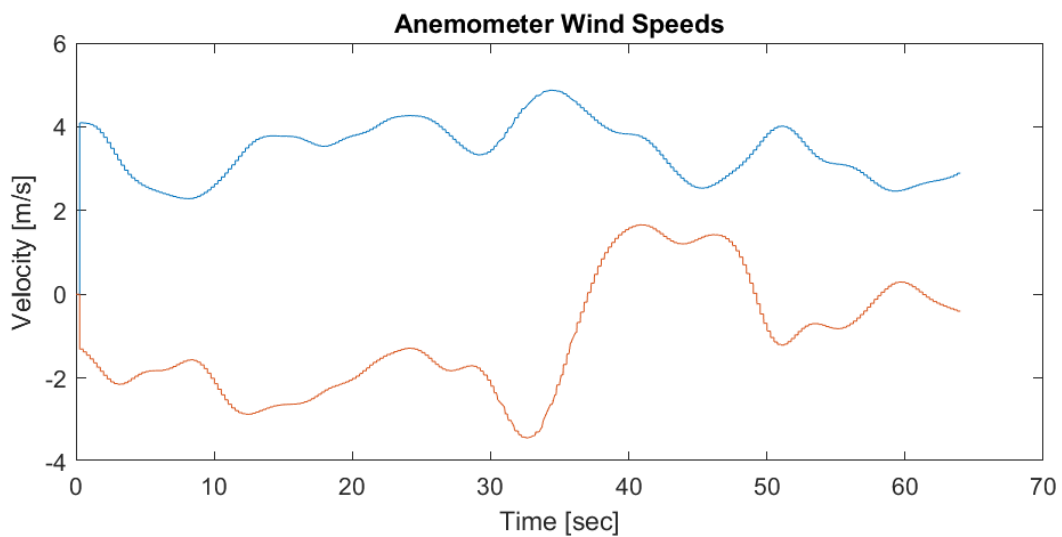
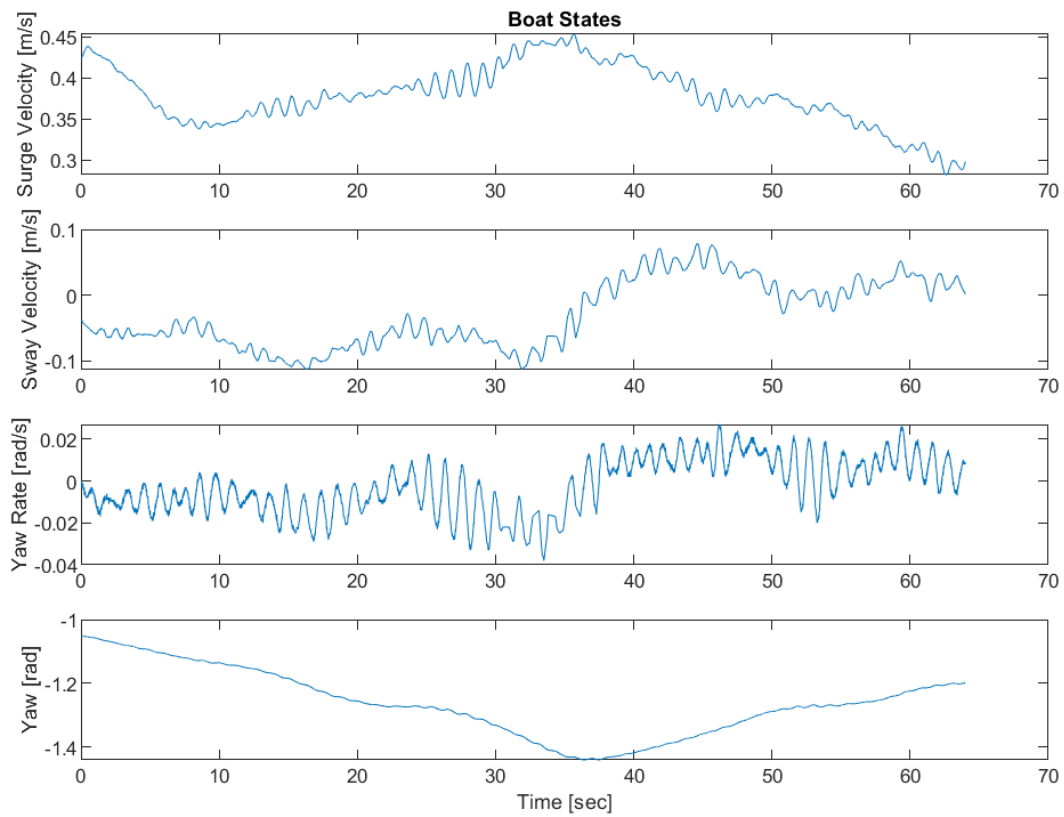
- [34] T. P. a. M. Blanke, "Mathematical Ship Modeling for Control Applications," The University of Newcastle.
- [35] D. L. L. Sicuro, "PHYSICALLY BASED MODELING AND SIMULATION OF A SHIP IN OPEN WATER 3-D VIRTUAL ENVIRONMENT," Naval Postgraduate School, 2003.
- [36] Ø. S. a. T. I. F. Roger Skjetne, "MODELING, IDENTIFICATION, AND ADAPTIVE MANEUVERING OF CYBERSHIP II: A COMPLETE DESIGN WITH EXPERIMENTS," *CAMS*, 2004.
- [37] B. C. S. Petr̃svec, "Dynamics-Aware Target Following for an Autonomous Surface Vehicle Operating under COLREGs in Civilian Traffic," *IEEE*, 2013.
- [38] I. R. B. K. D. v. E. a. M. R. D. Wilhelm B. Klinger, "Control of an Unmanned Surface Vehicle with Uncertain Displacement and Drag," *IEEE*, 2016.
- [39] F. Jenne, "Simulation and Control Optimization of an autonomous sail boat," Swiss Federal Institute of Technology Zurich, 2010.
- [40] J. Carlton, *Marine Propellers and Propulsion*, Waltham: Elsevier Ltd., 2010.
- [41] N. A. Manzini, "USV Path Planning Using Potential Field Model," Naval postgraduate School, Monterey, 2017.
- [42] S. Abdel-latif, M. Abdel-geliel and E. E. Zakzouk, "Simulation of Ship Maneuvering Behavior Based on The Modular Mathematical Model," in *Mediterranean Conference on Control & Automation (MED)*, Crete, 2013.

Appendix

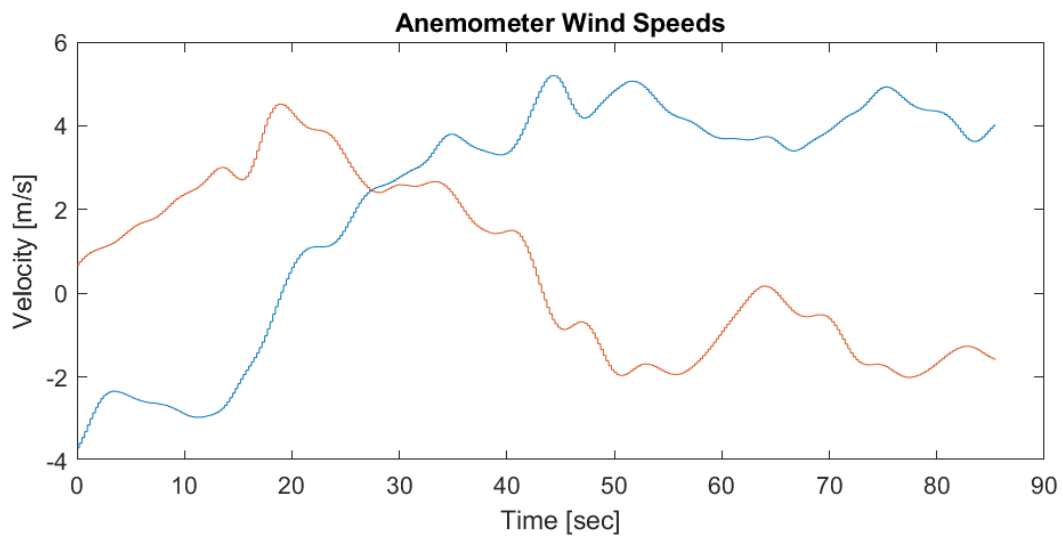
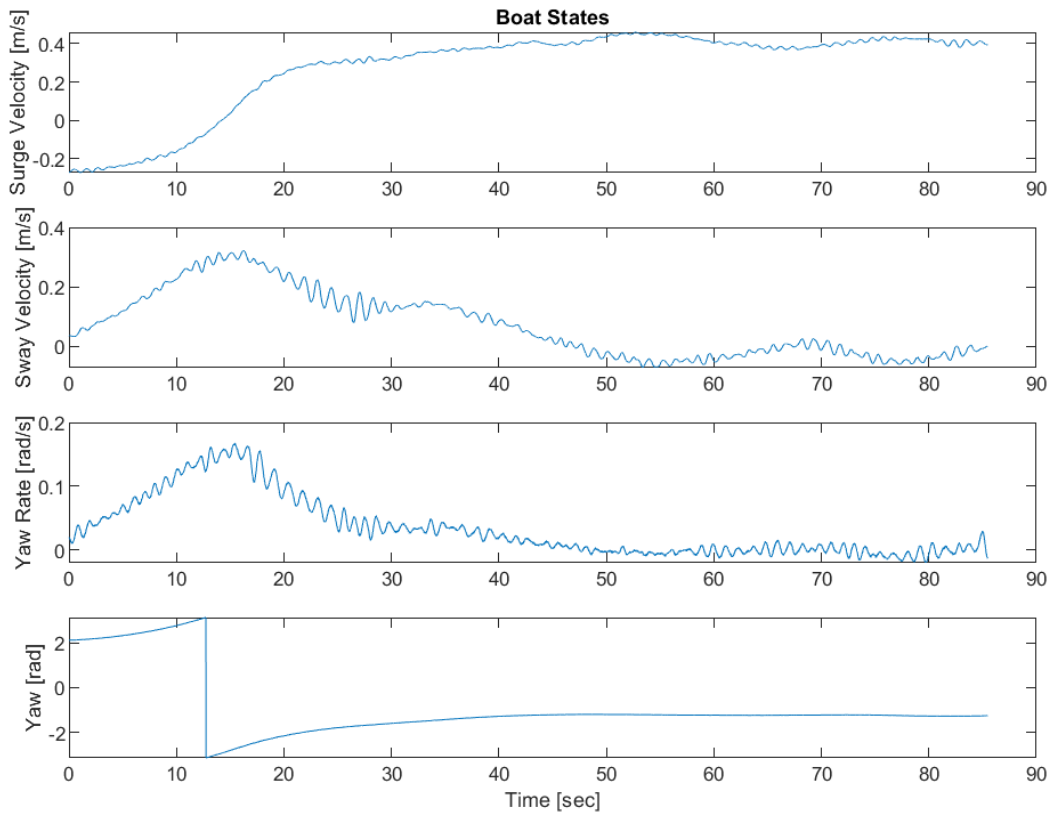
Data Test Sets

1.1. Wind Disturbance Data Sets

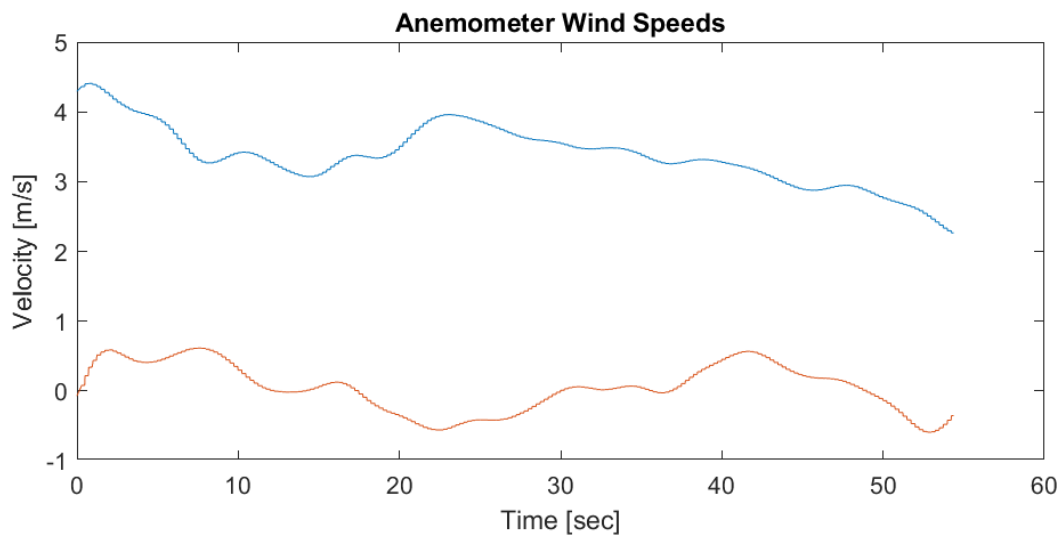
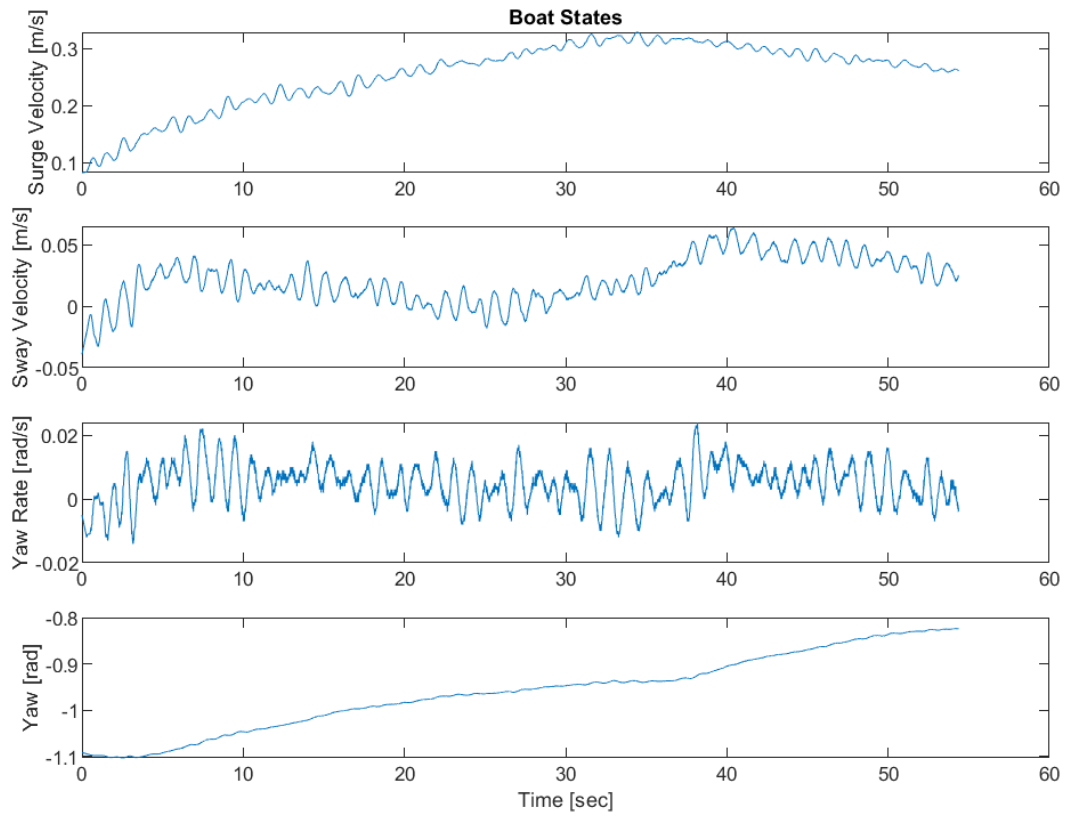
1.1.1. Wind Set 1



1.1.2. Wind Set 2

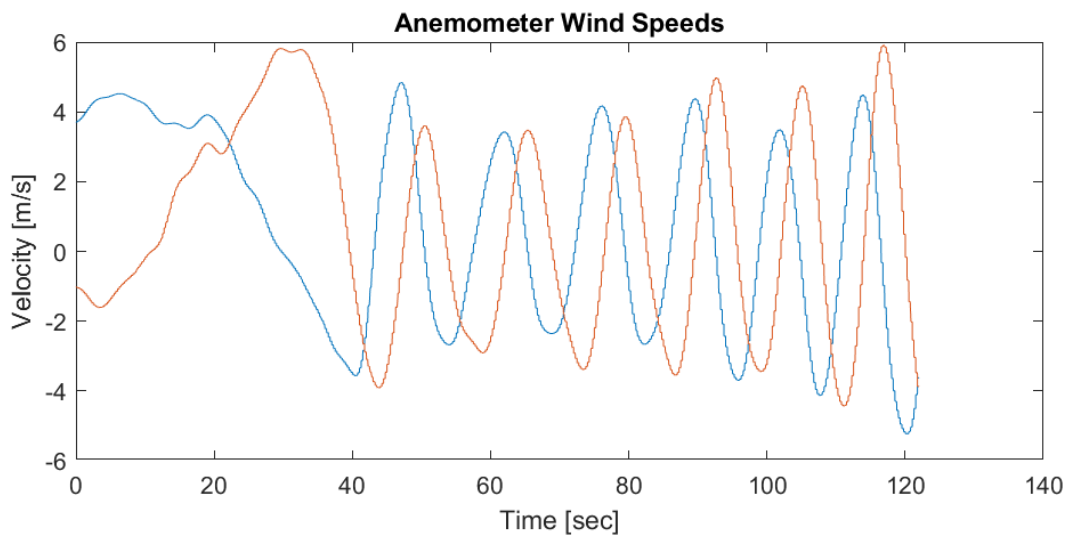
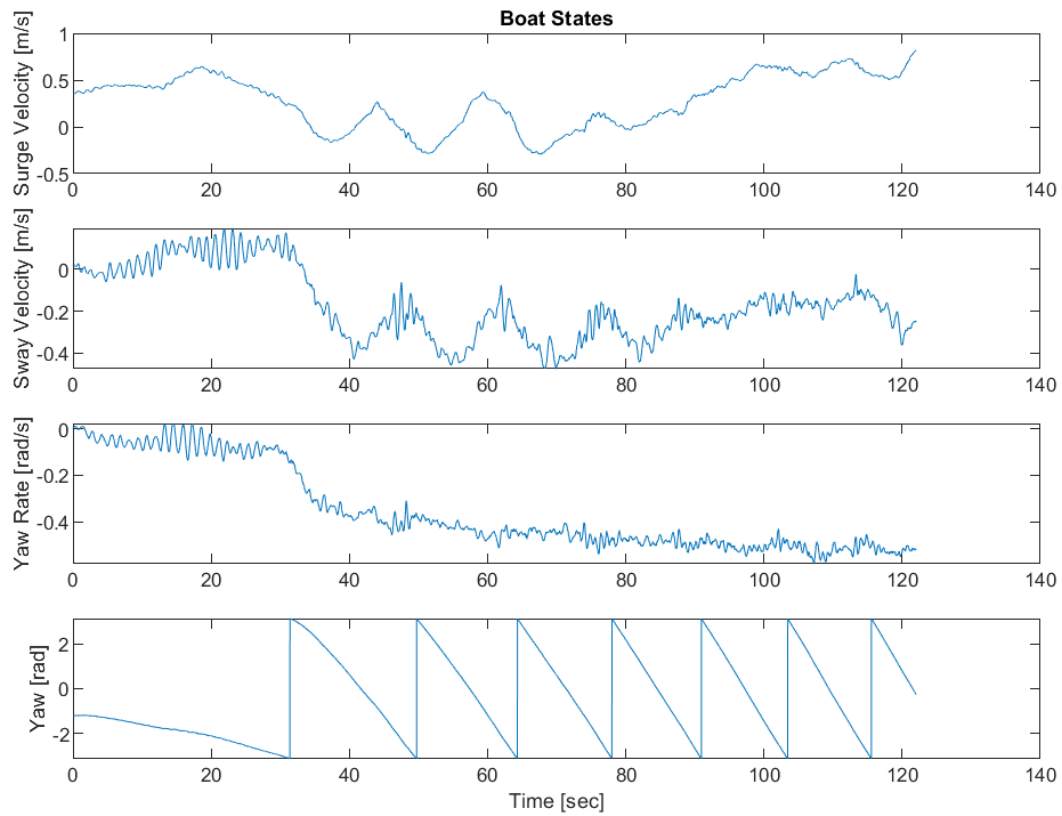


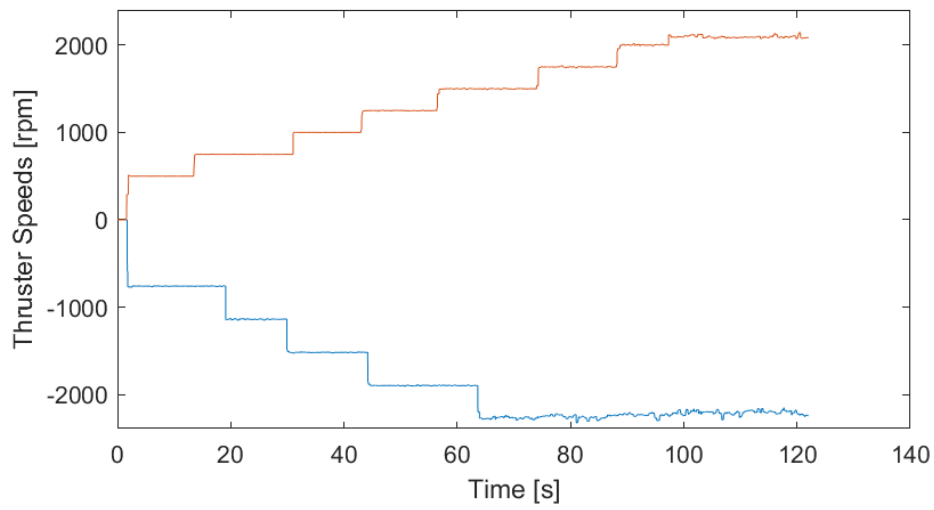
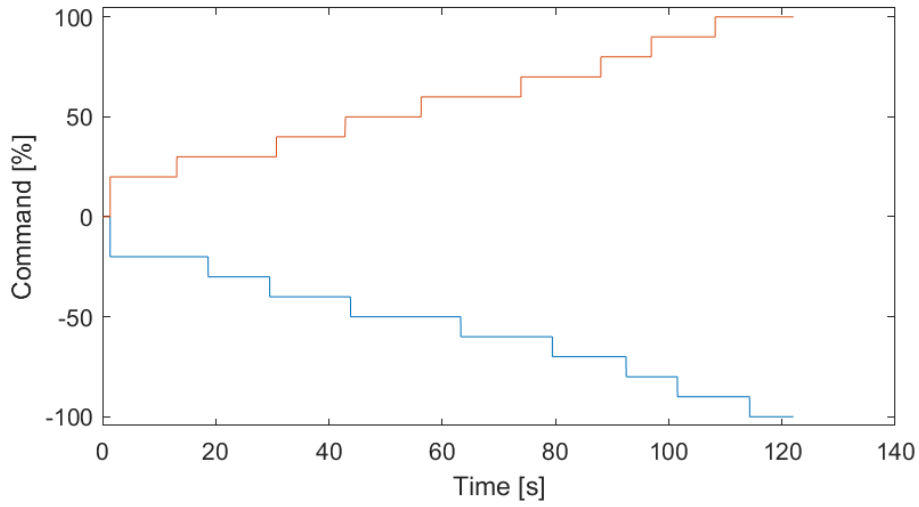
1.1.3. Wind Set 3



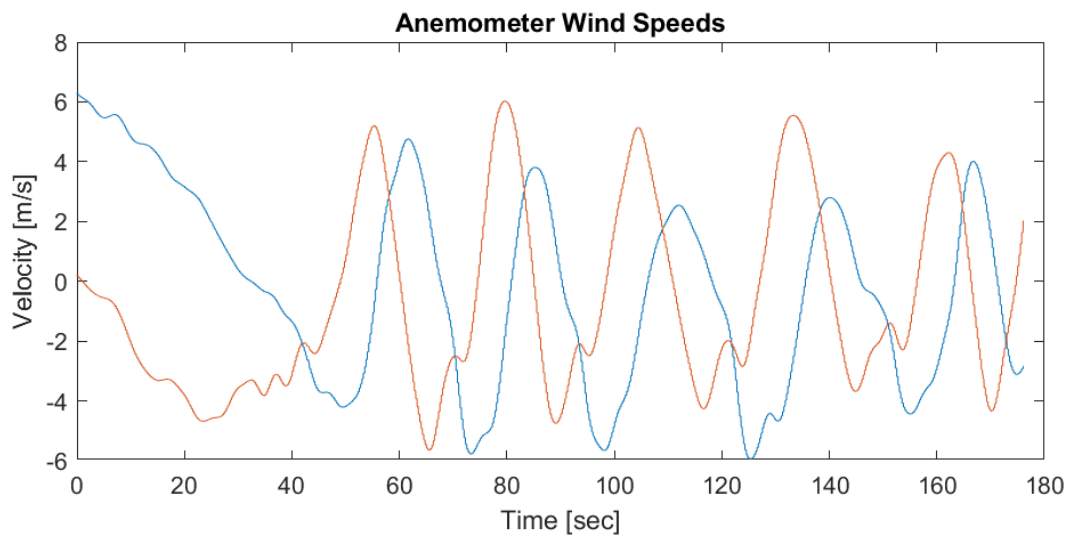
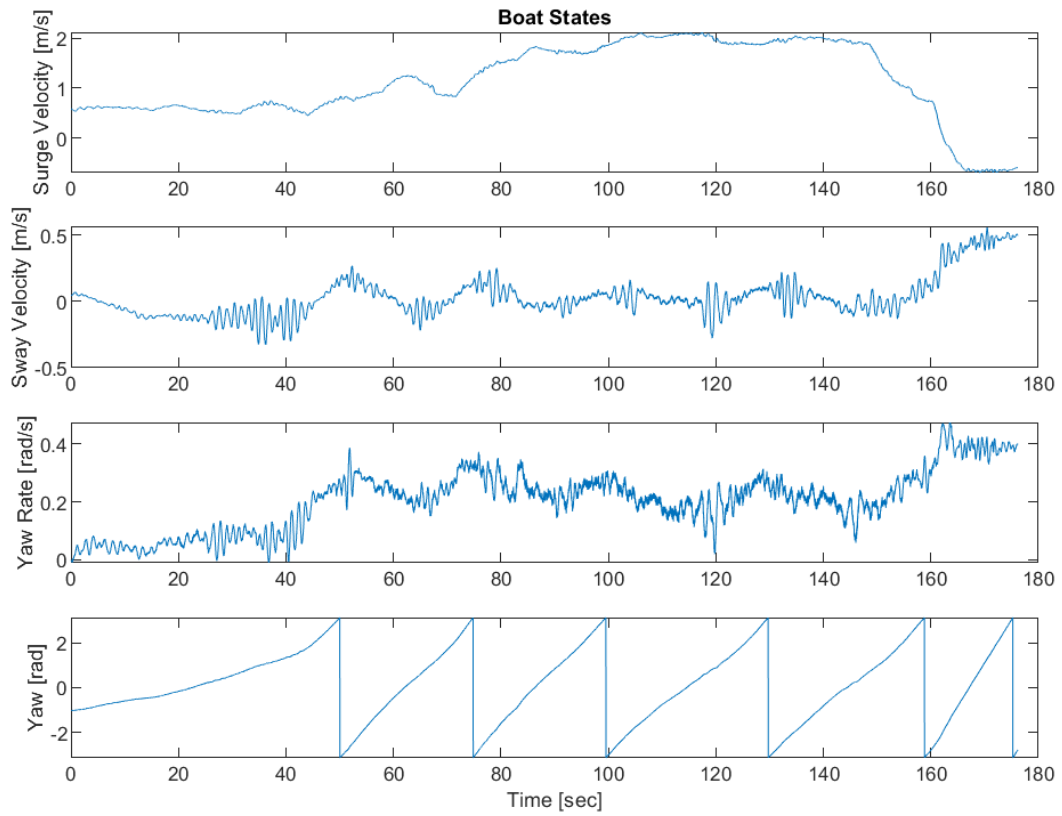
1.2. Spin Test Data Sets

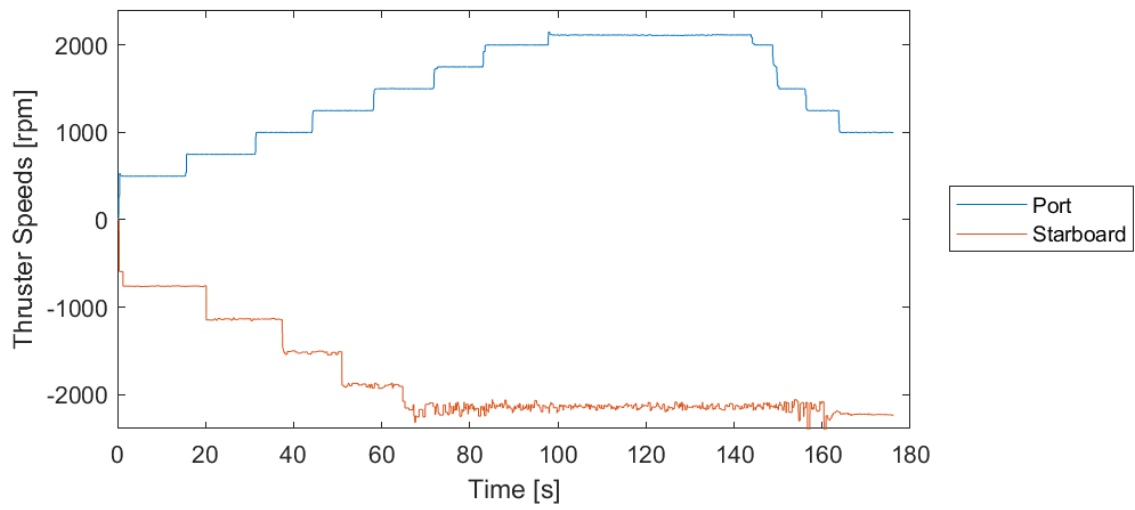
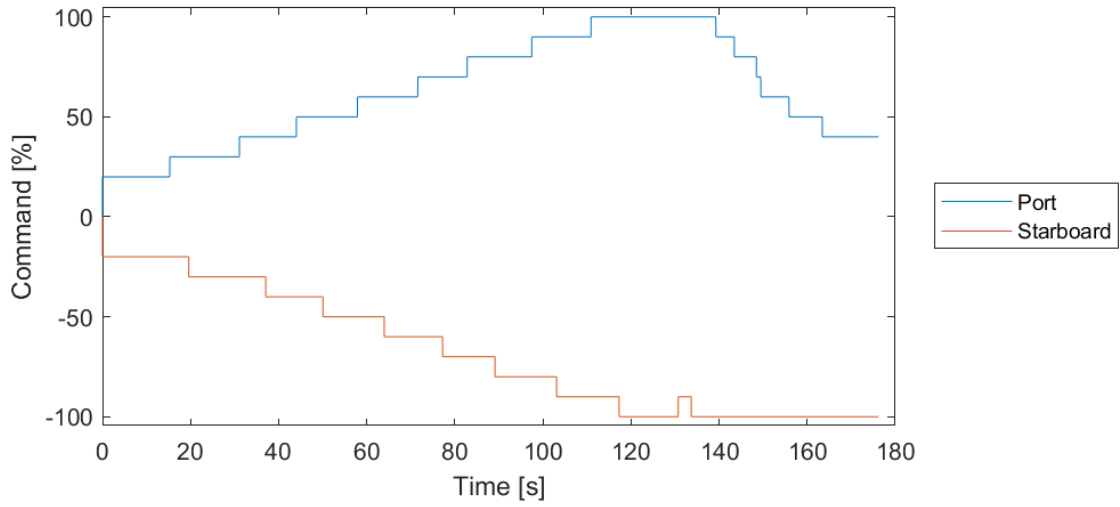
1.2.1. Spin Set 1





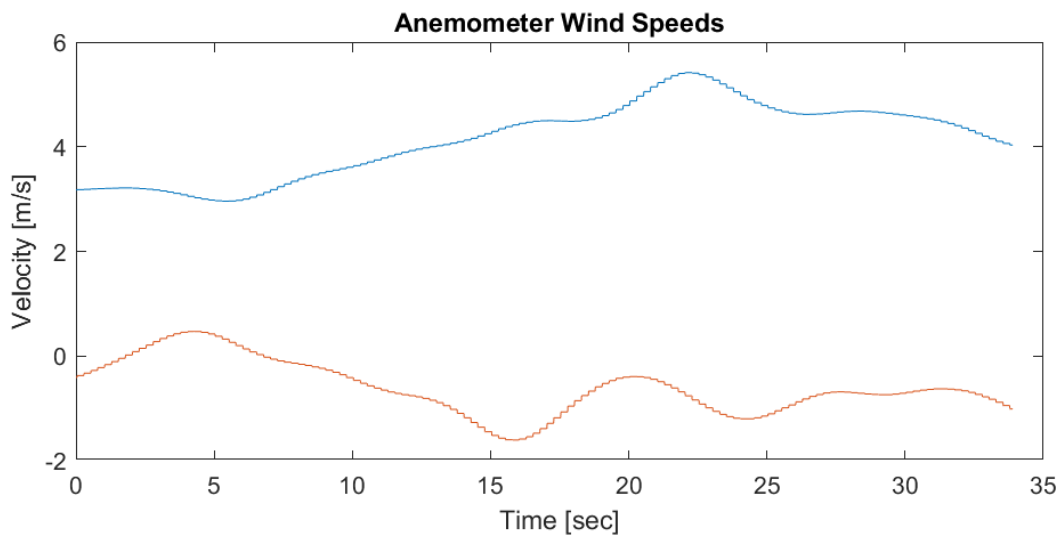
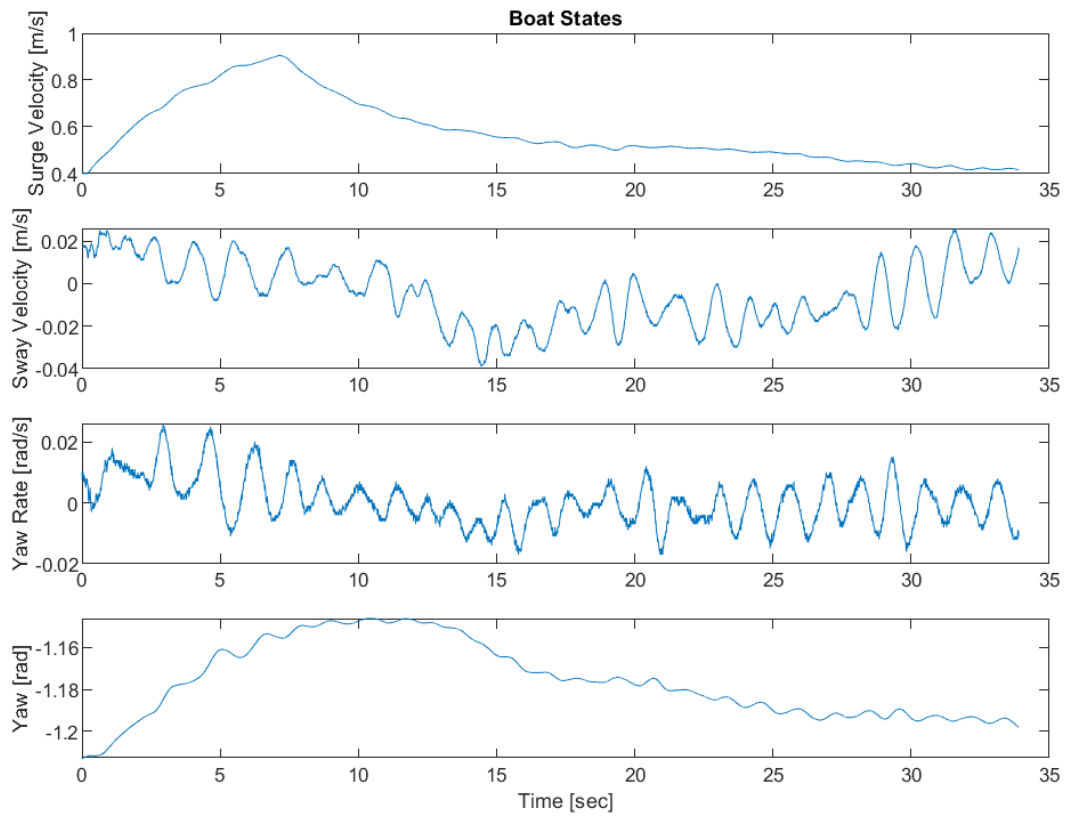
1.2.2. Spin Set 2

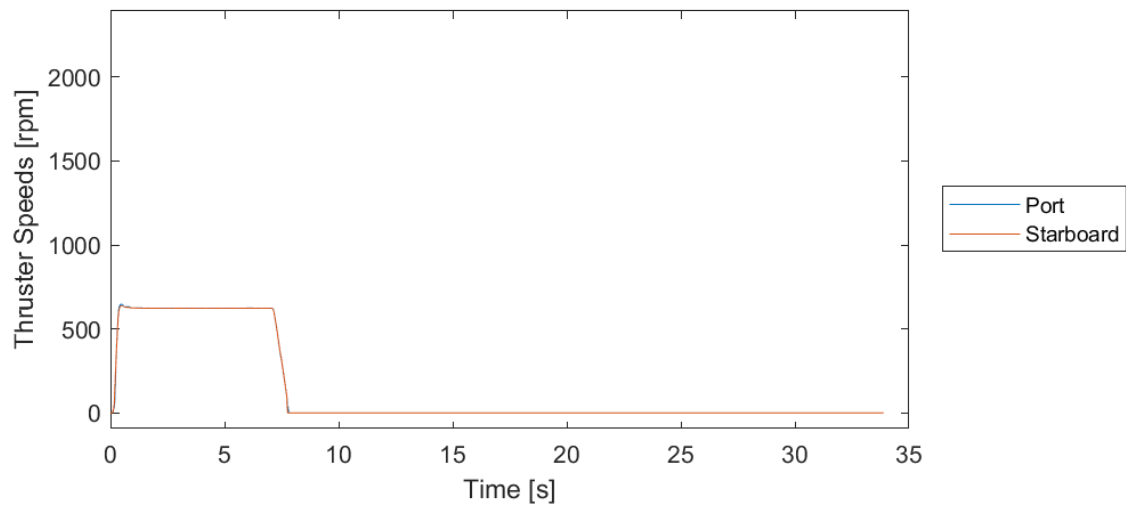
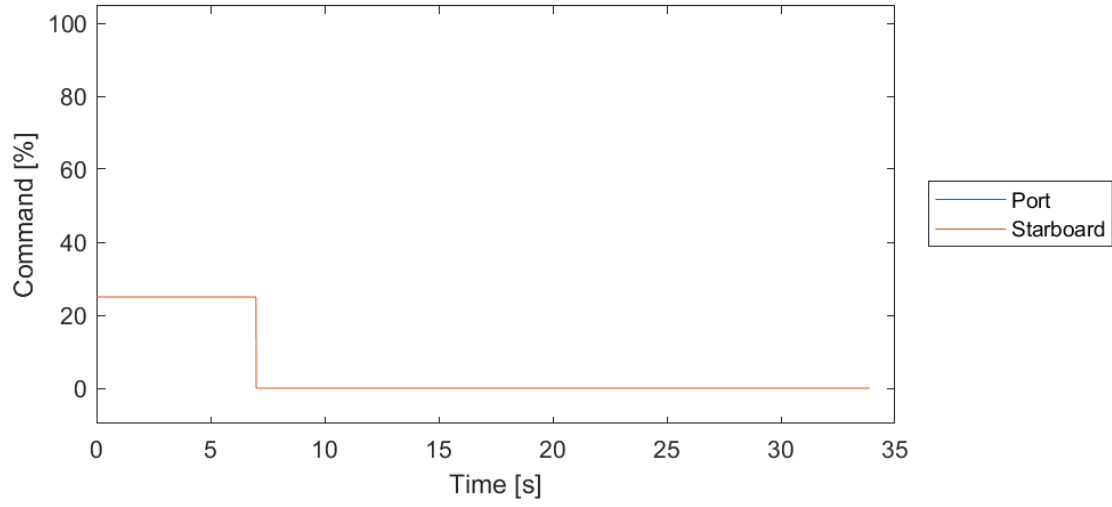




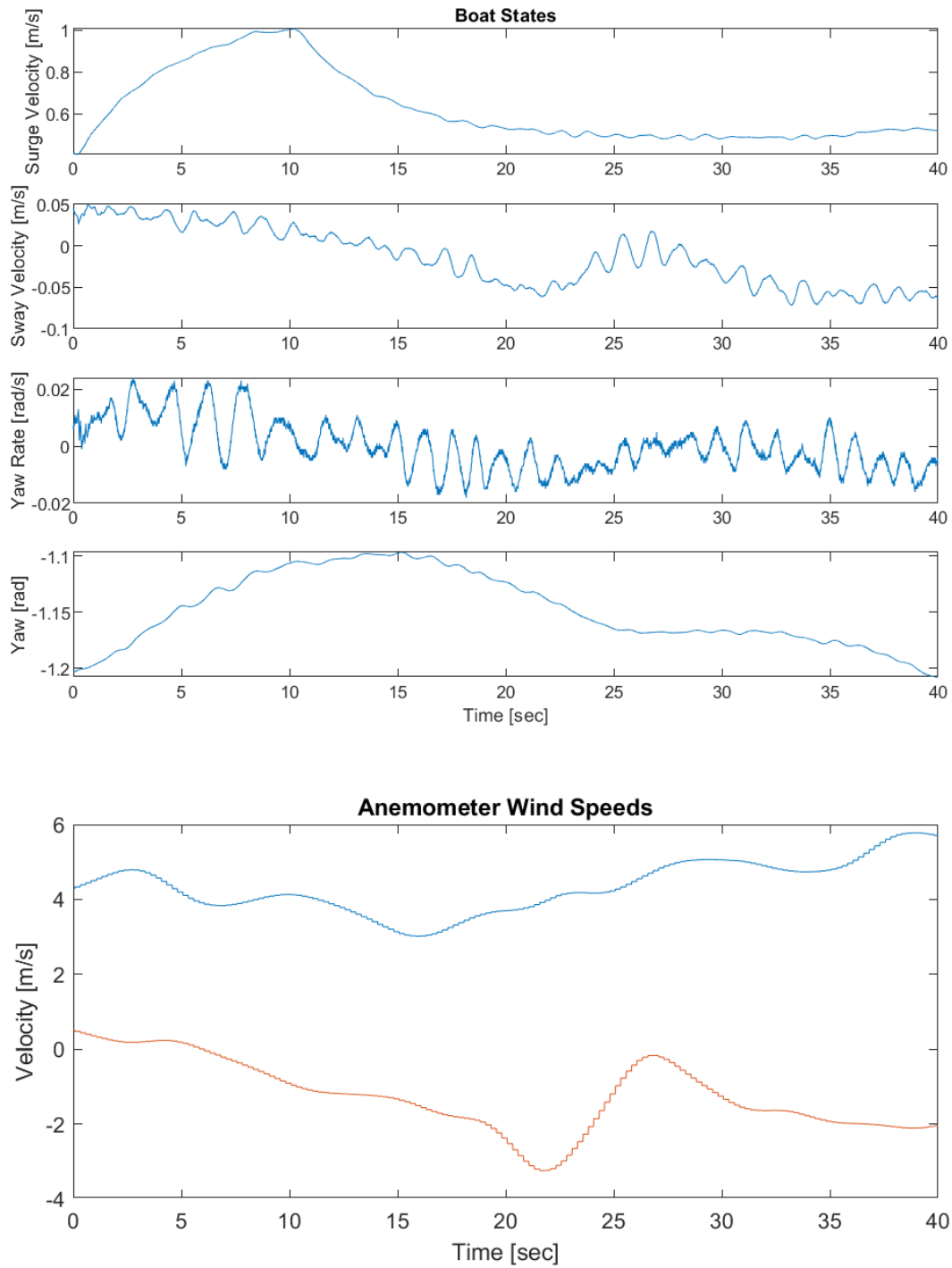
1.3. Stopping Test Data Sets

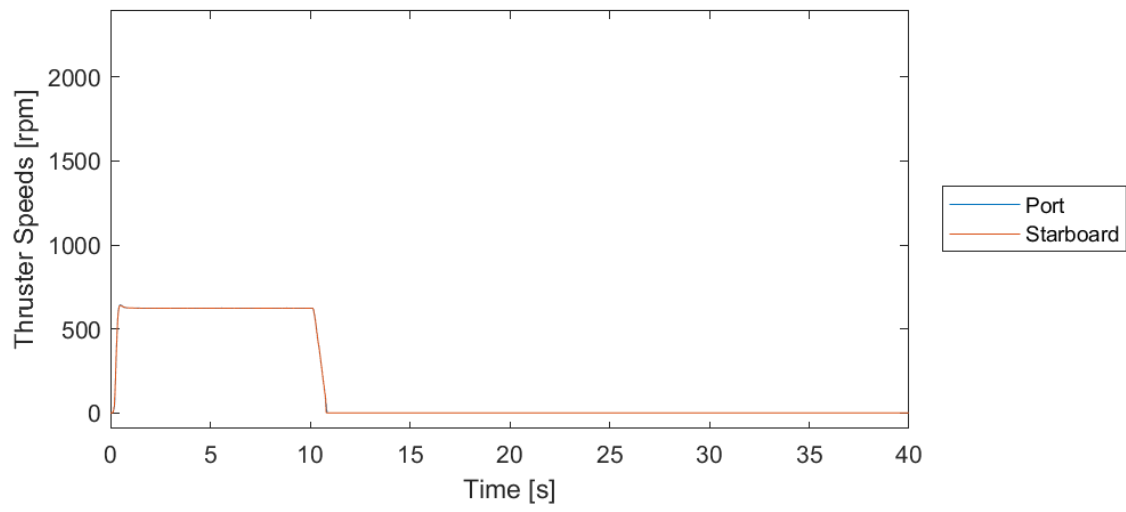
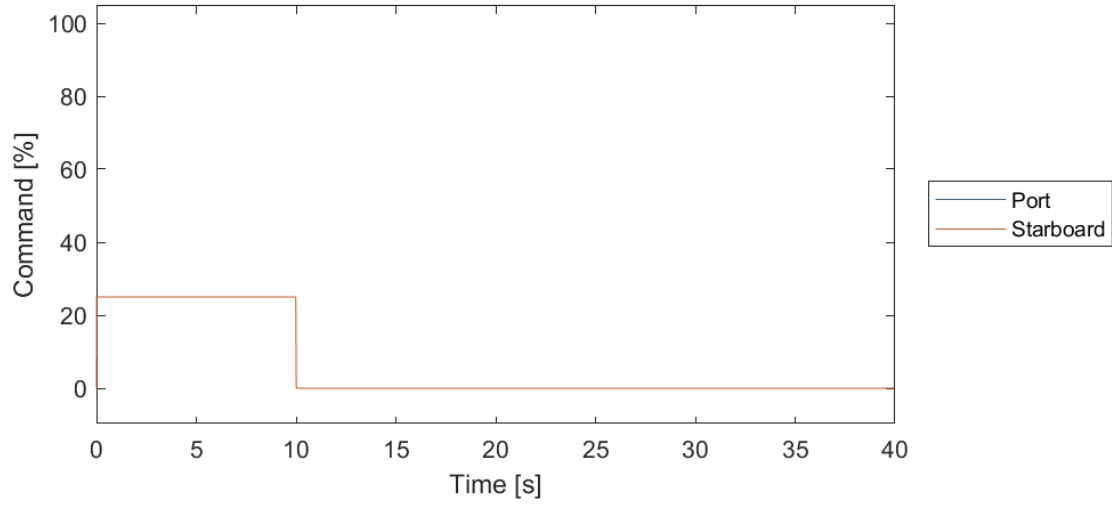
1.3.1. Stopping Set 1



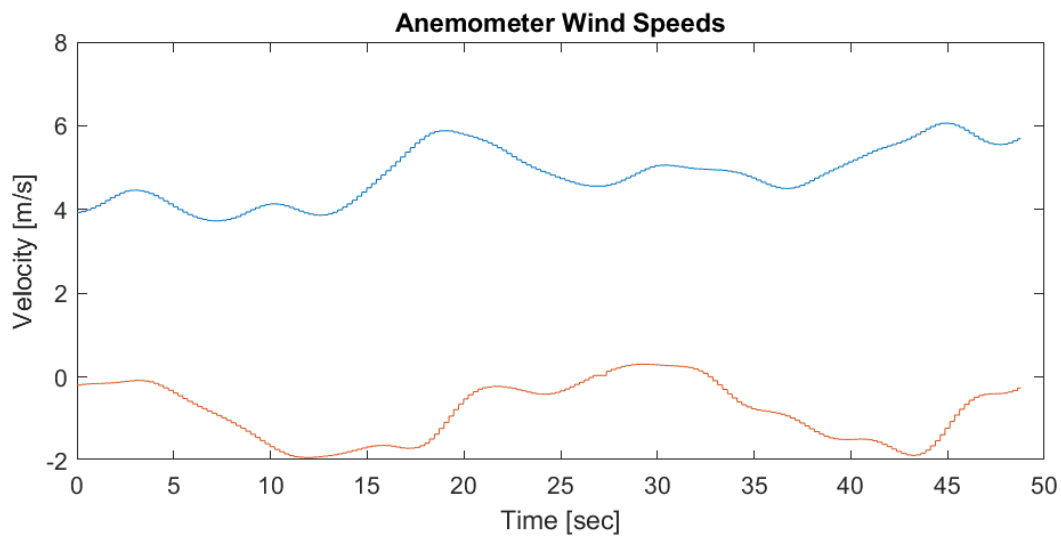
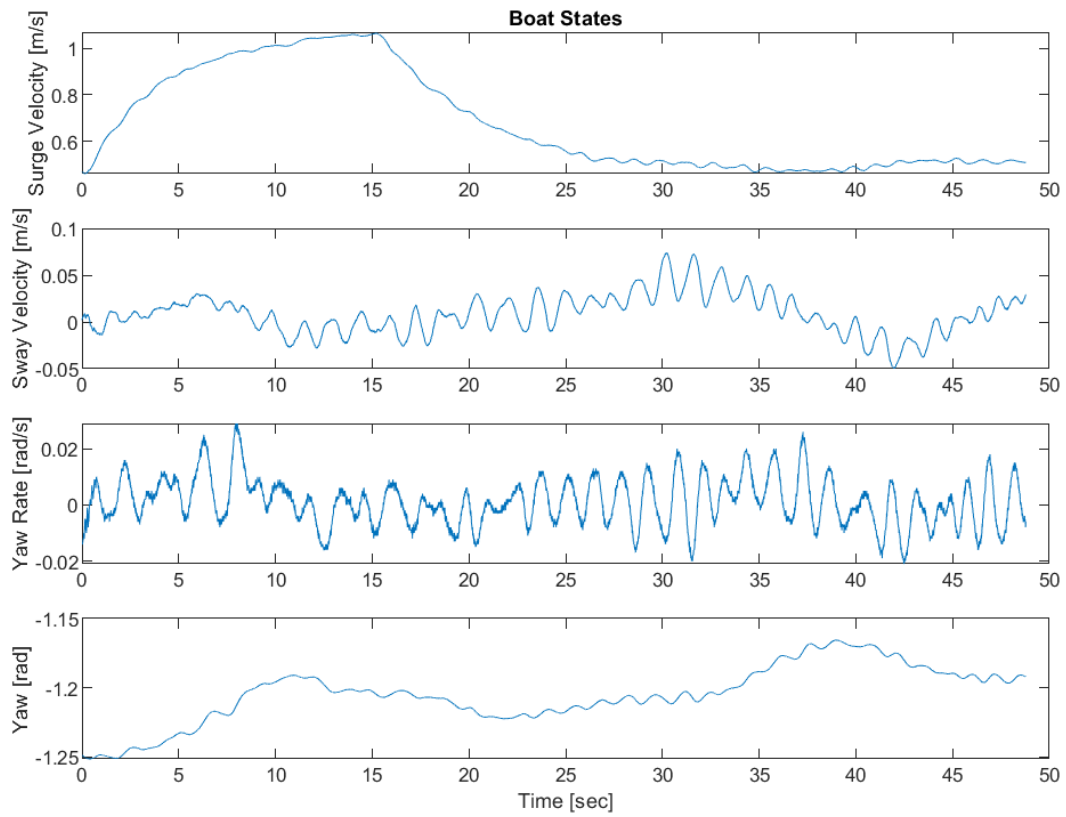


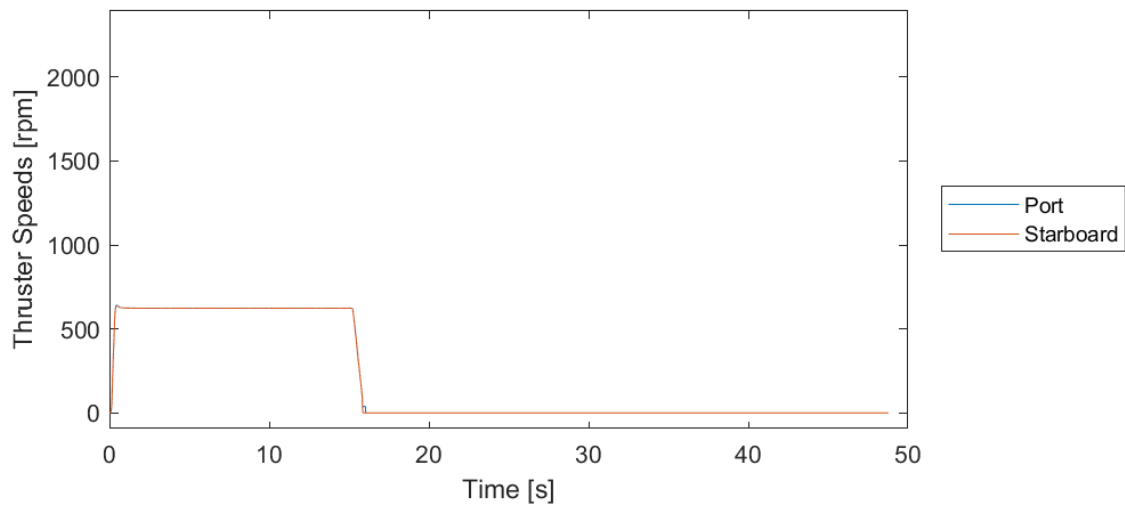
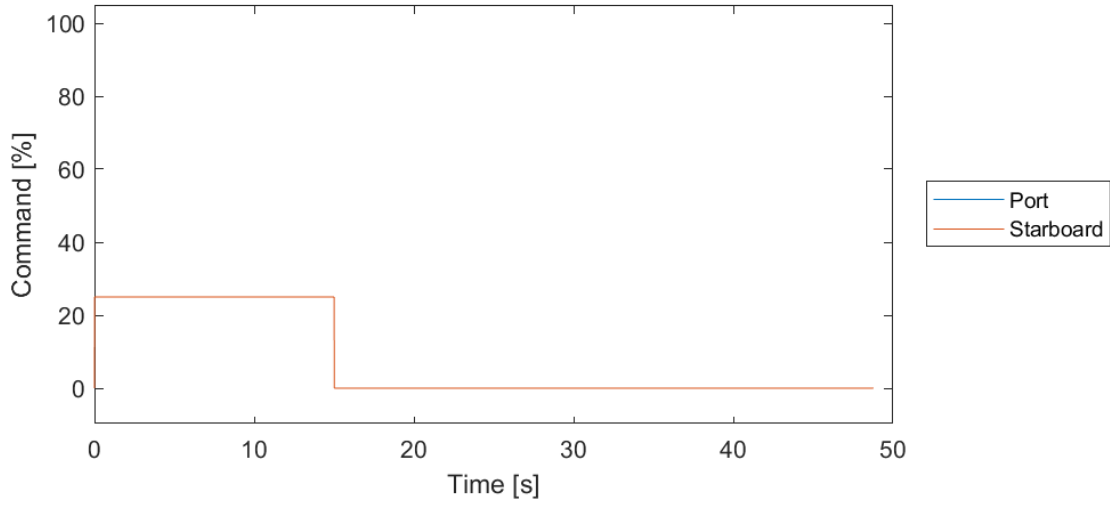
1.3.2. Stopping Set 2



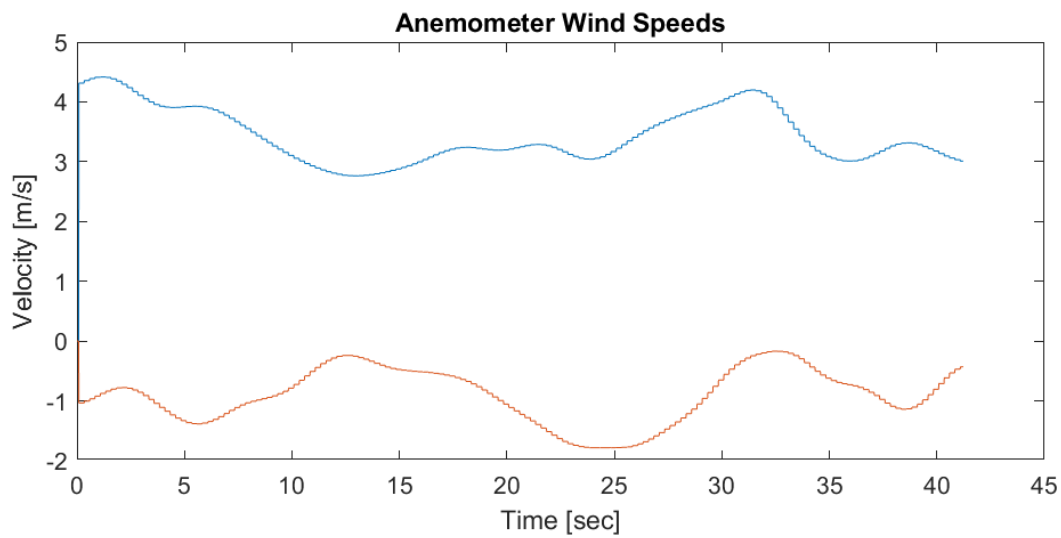
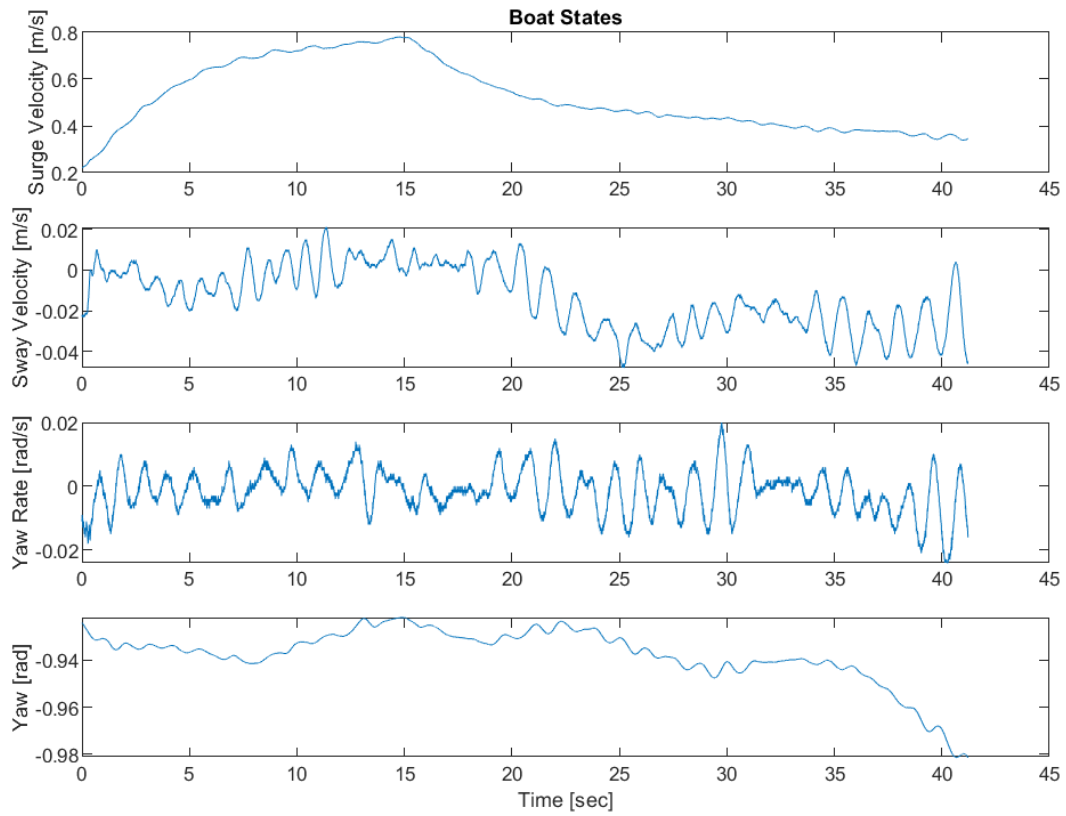


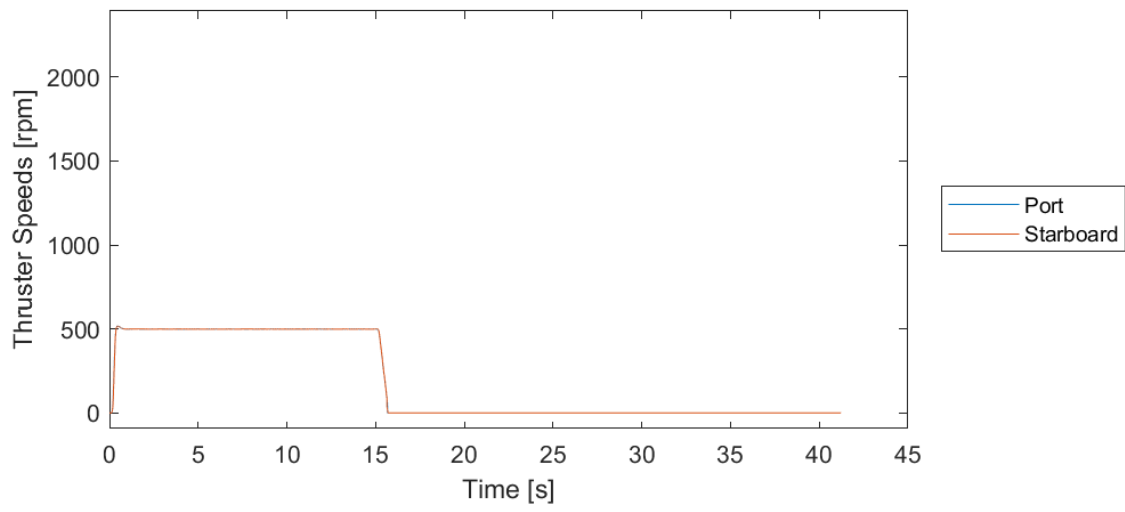
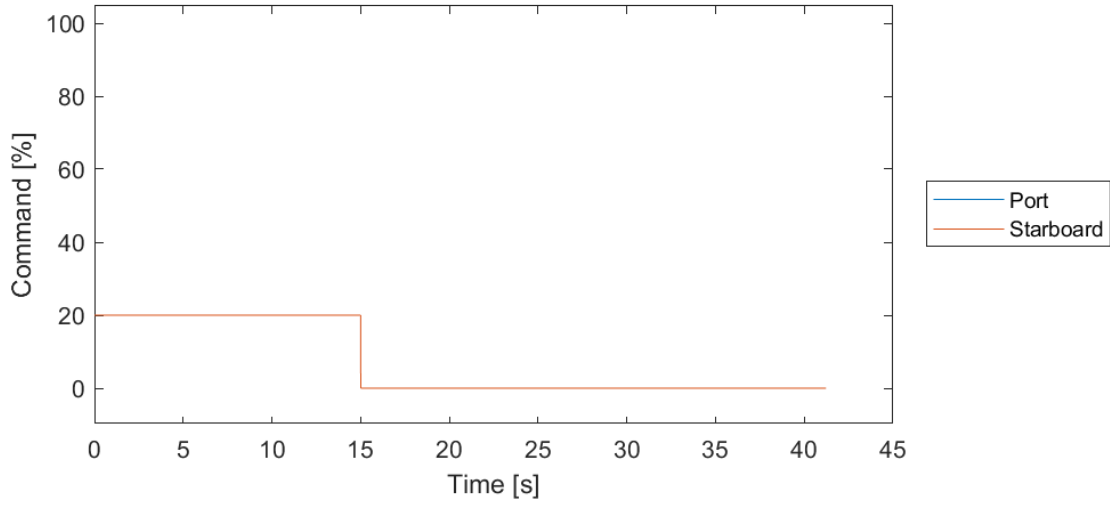
1.3.3. Stopping Set 3



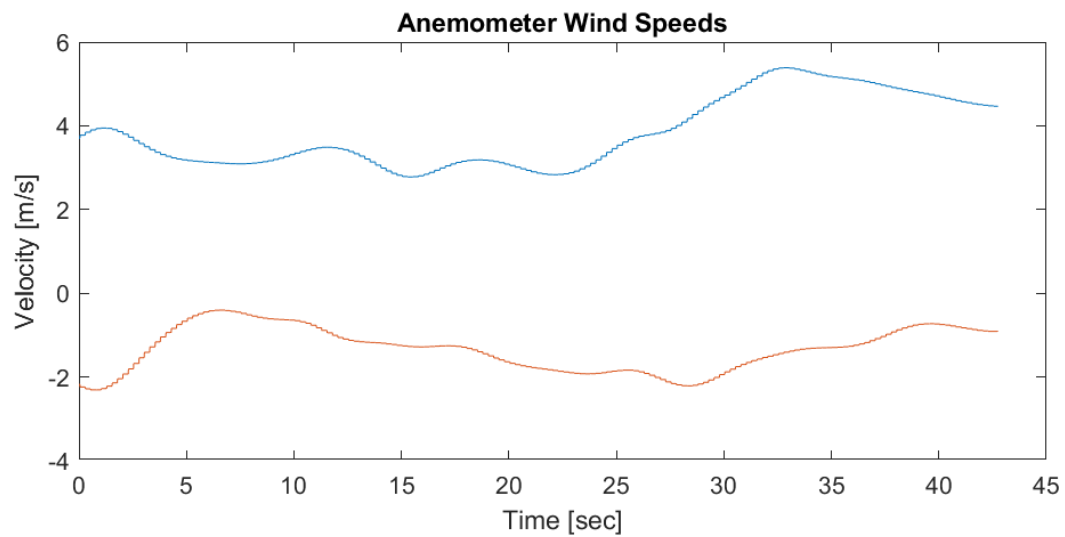
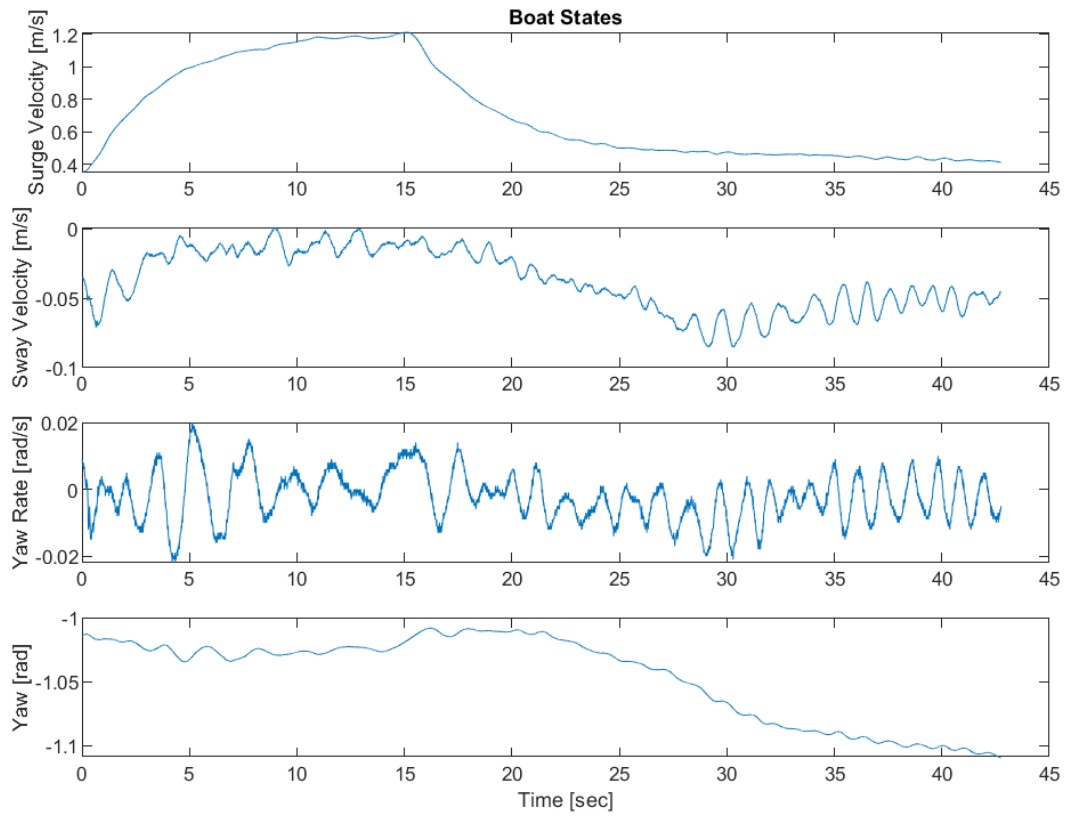


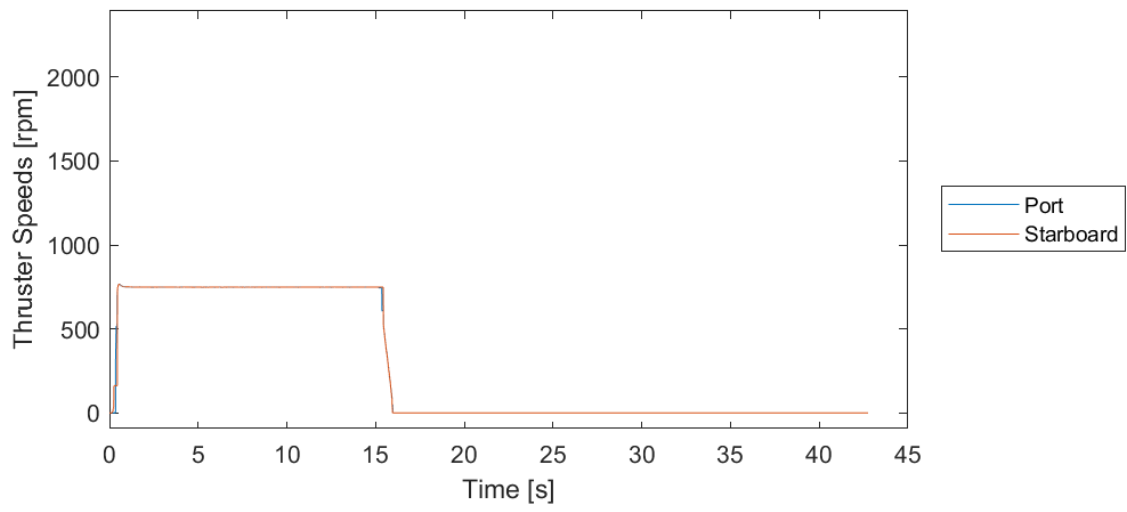
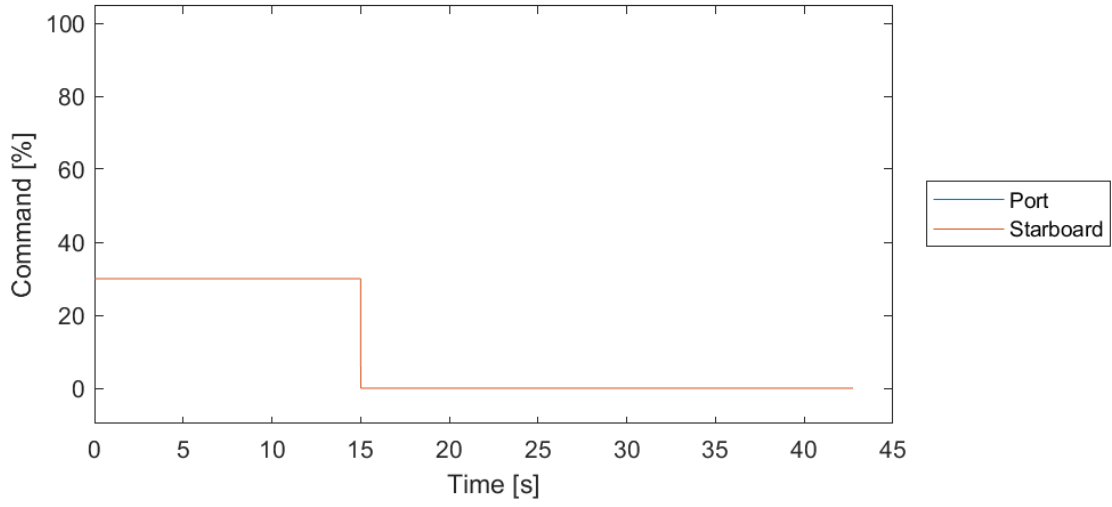
1.3.4. Stopping Set 4



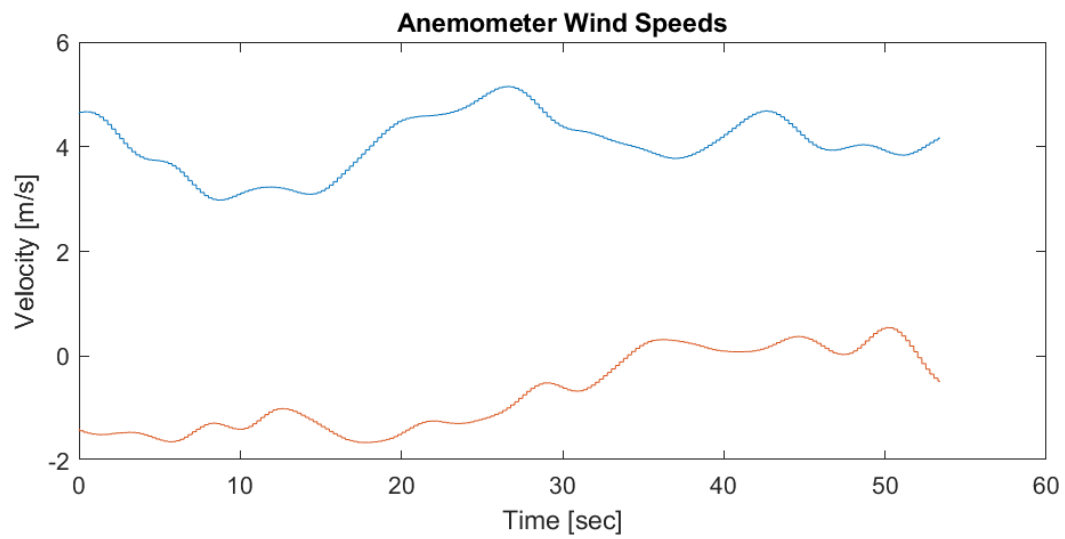
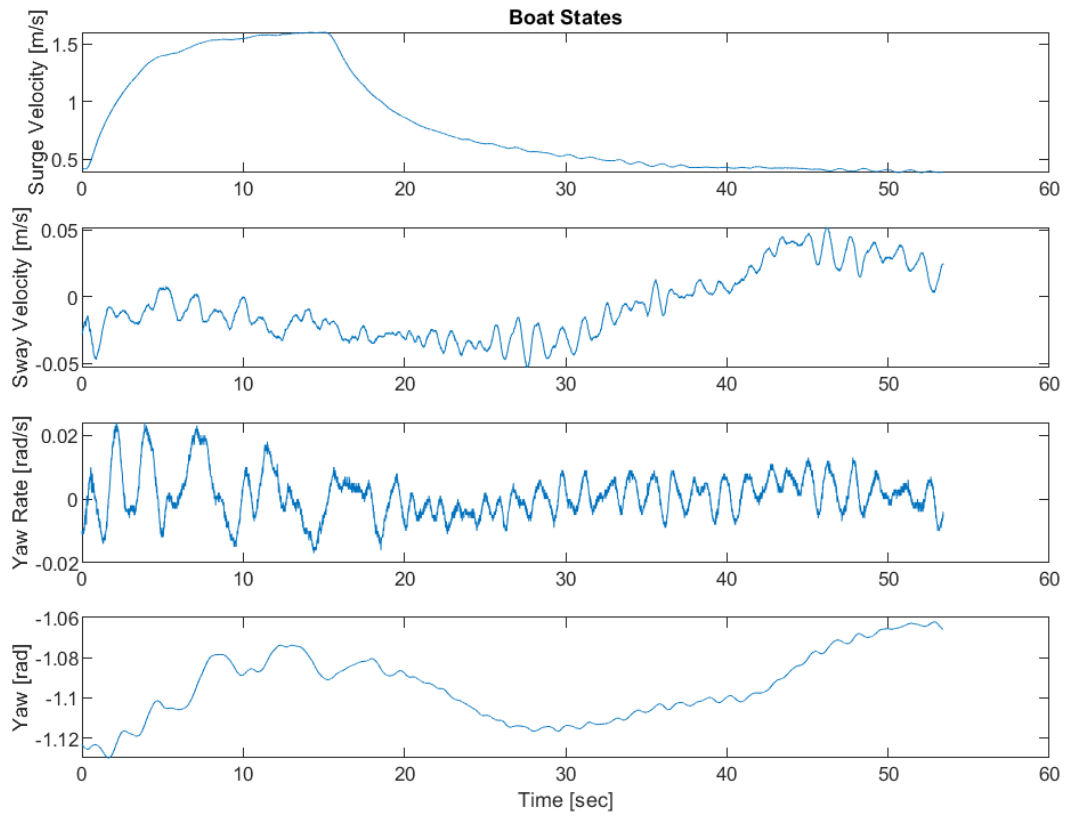


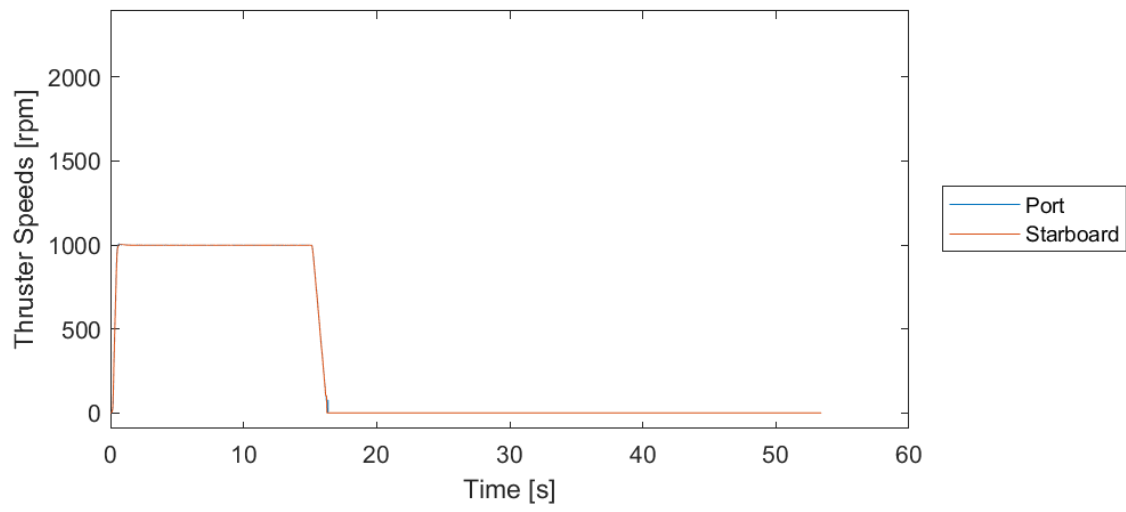
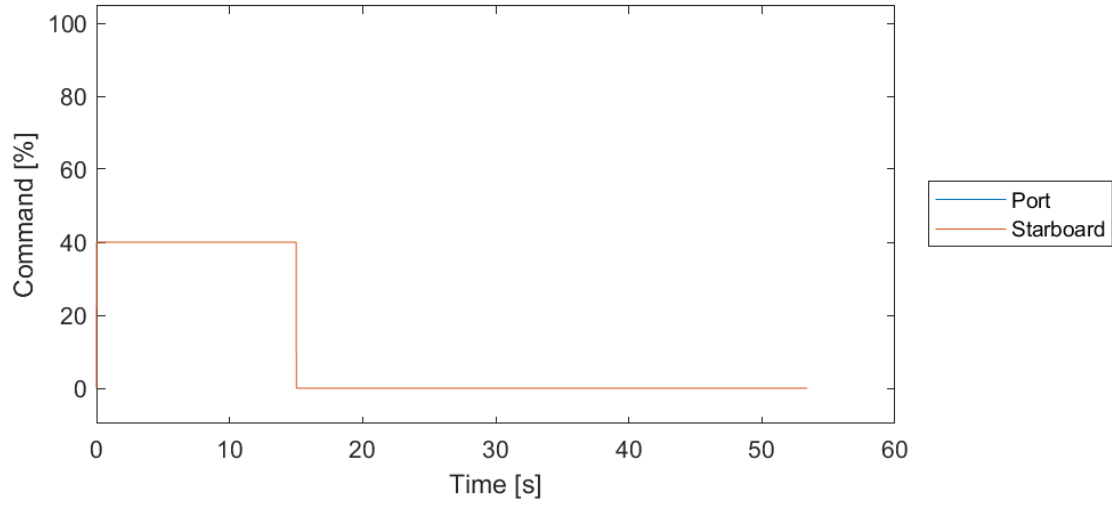
1.3.5. Stopping Set 5



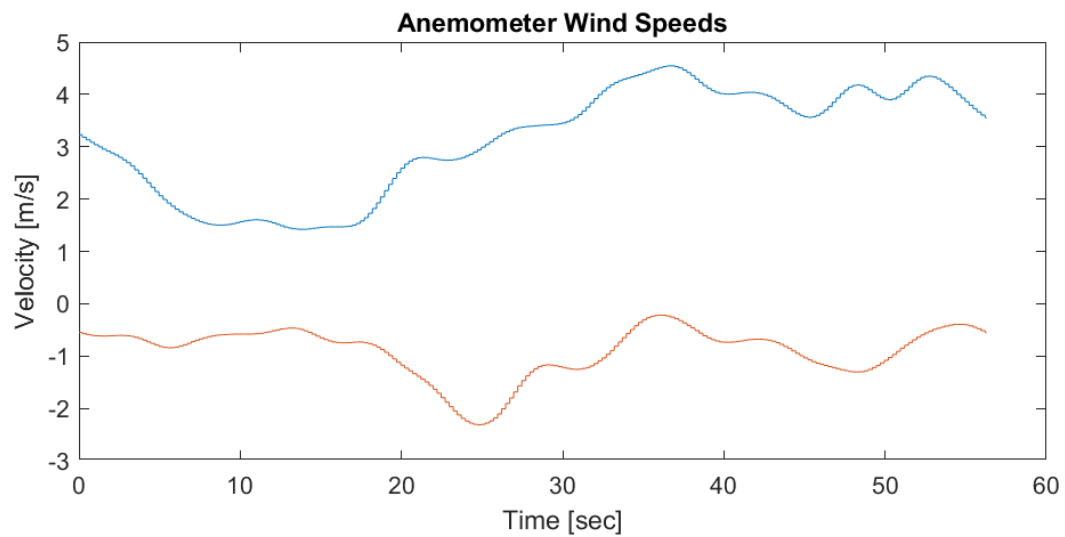
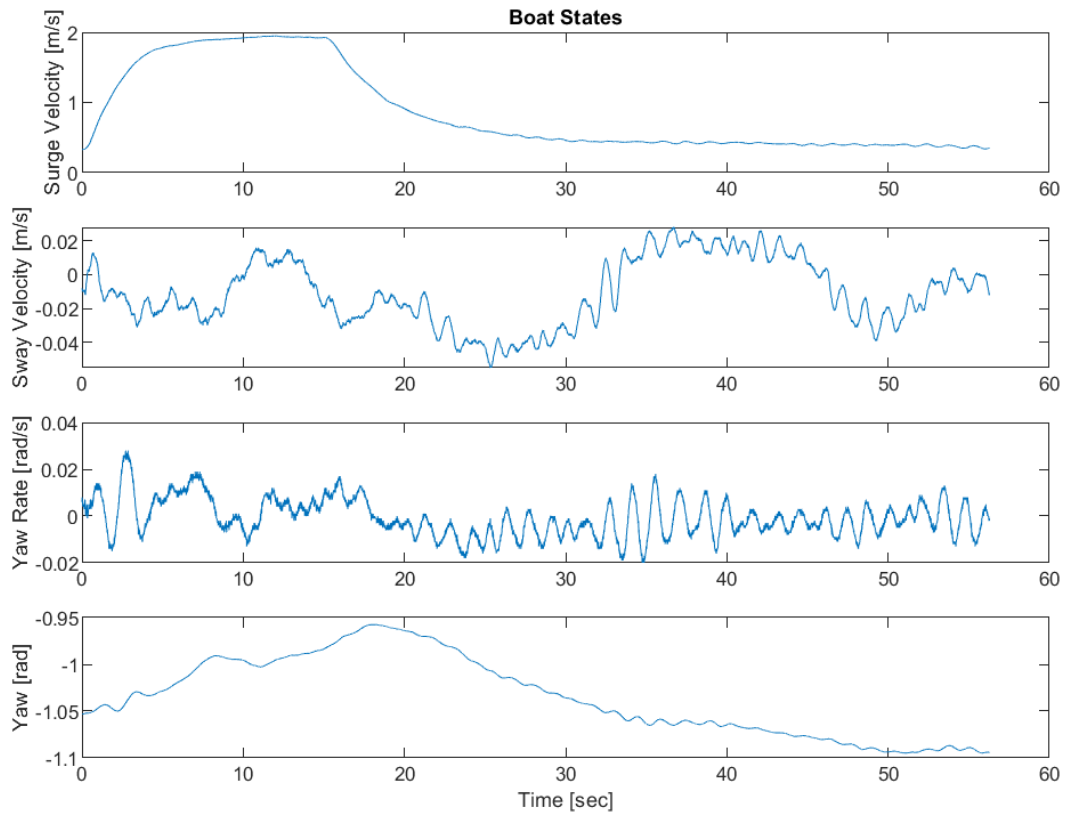


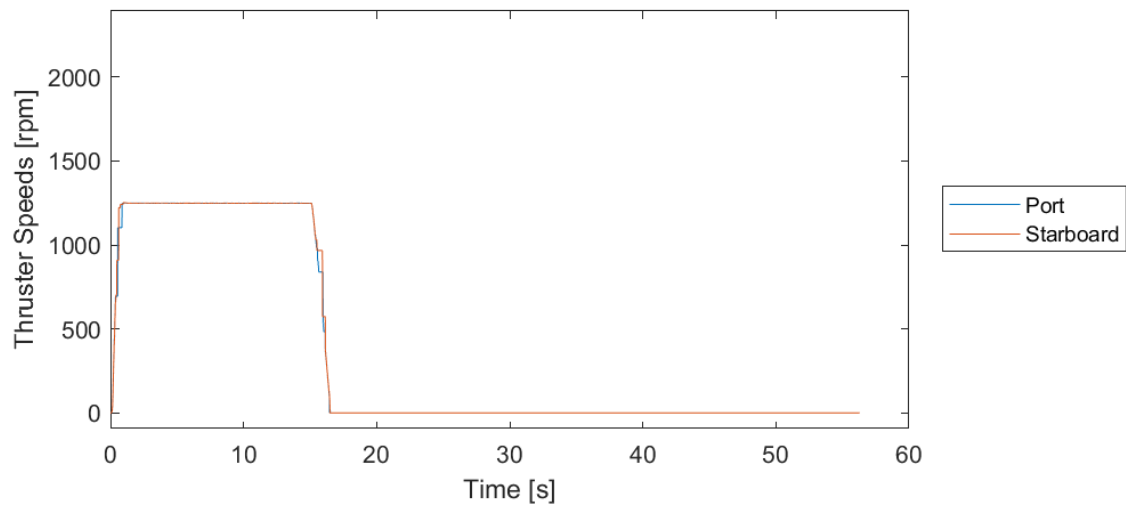
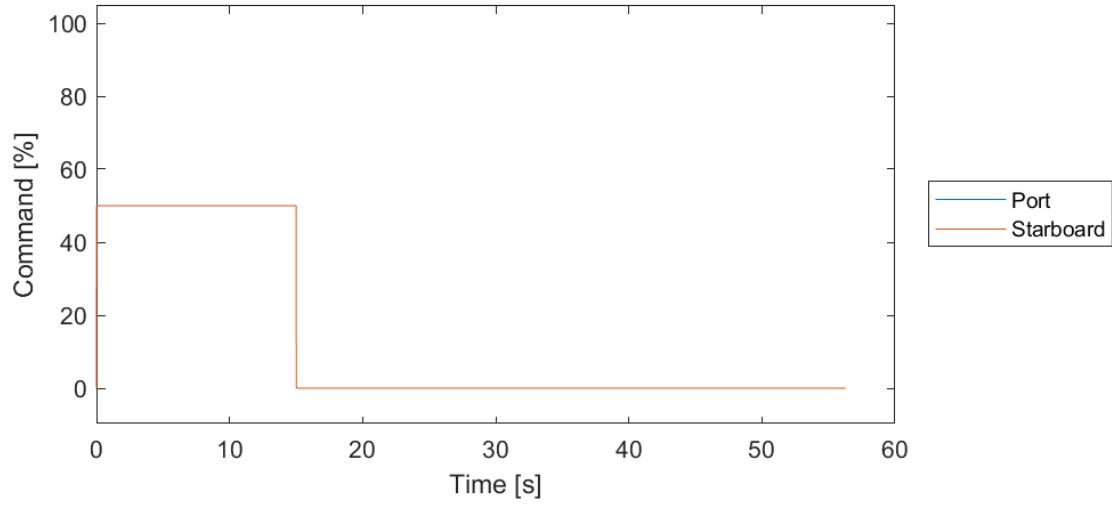
1.3.6. Stopping Set 6



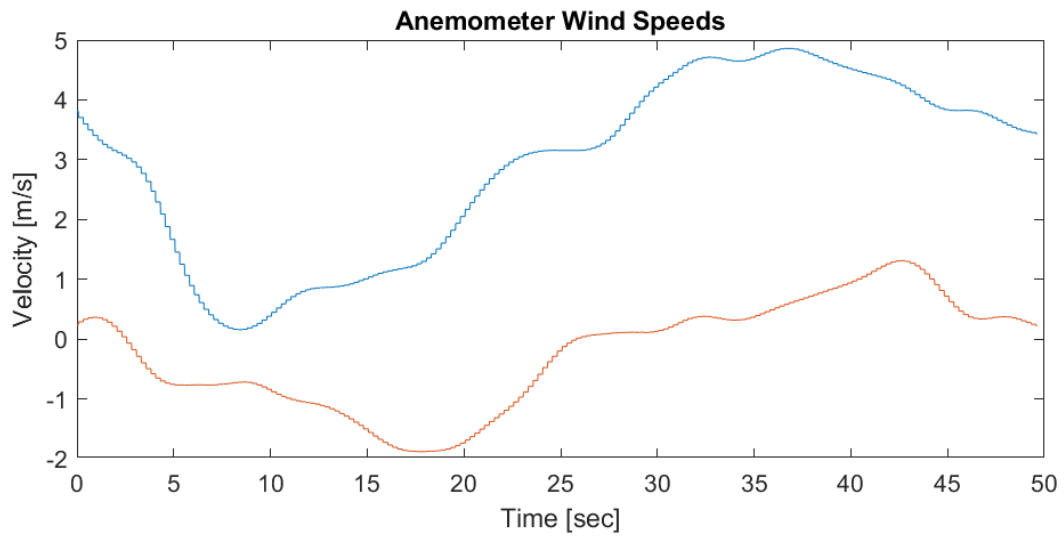
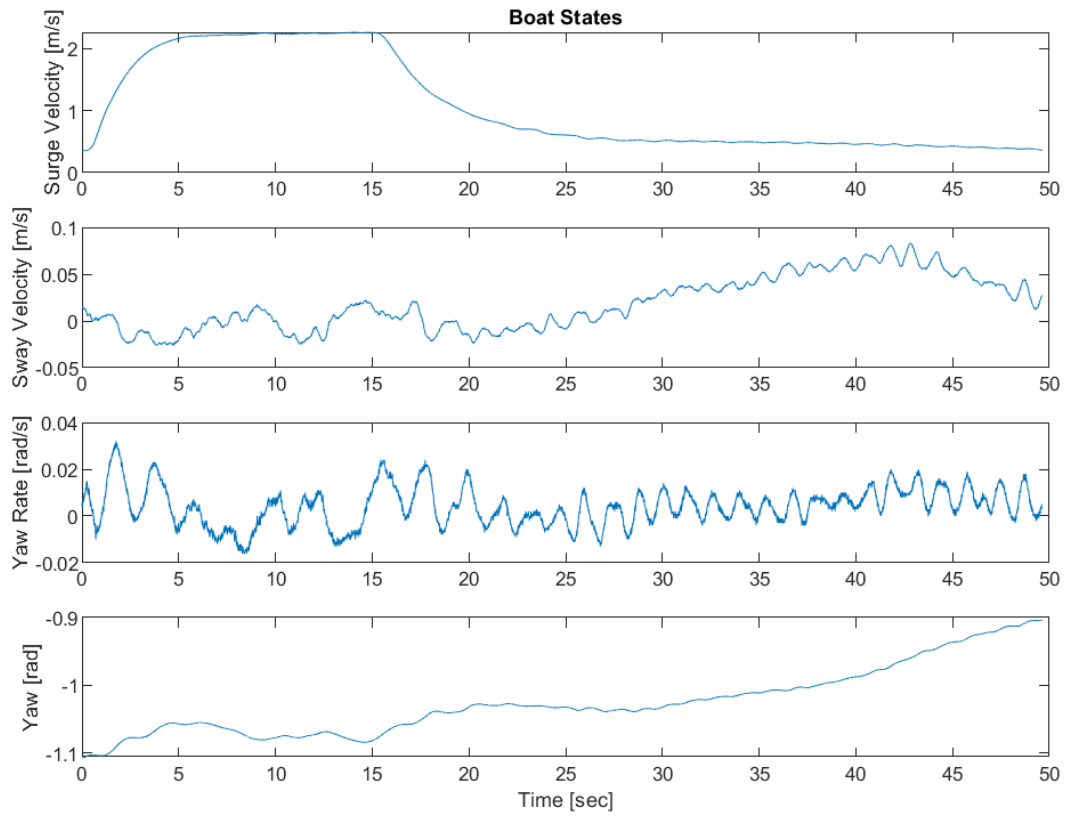


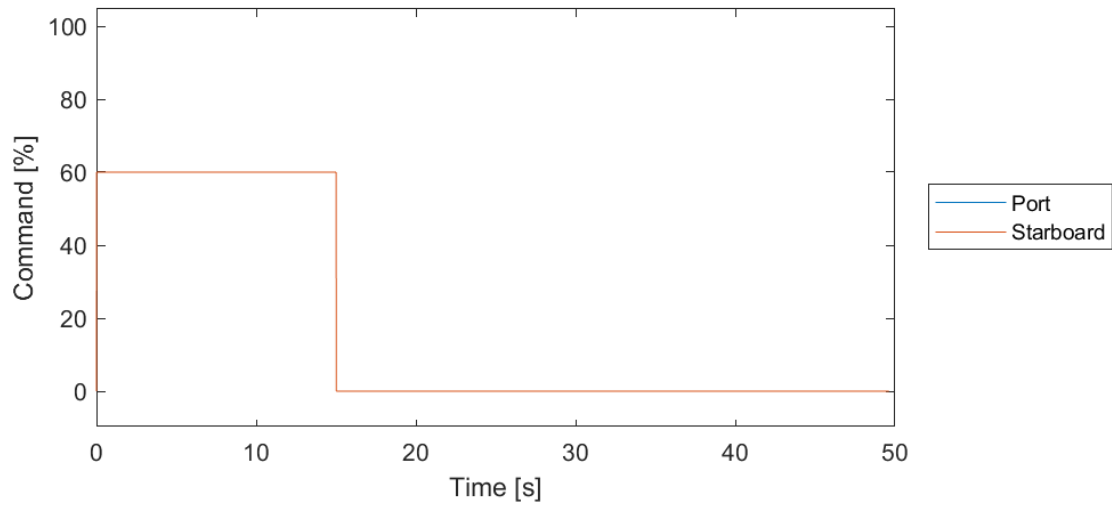
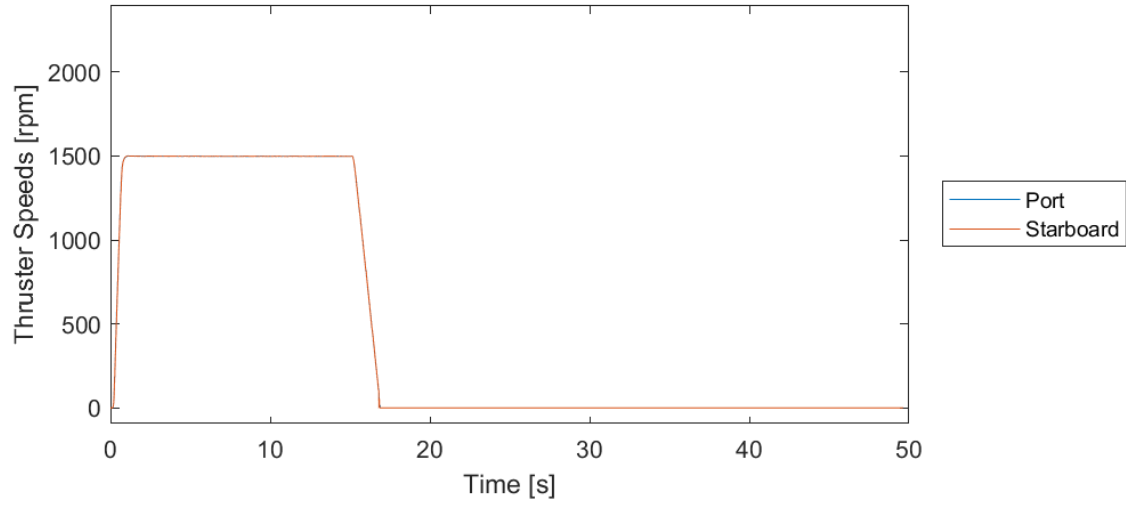
1.3.7. Stopping Set 7



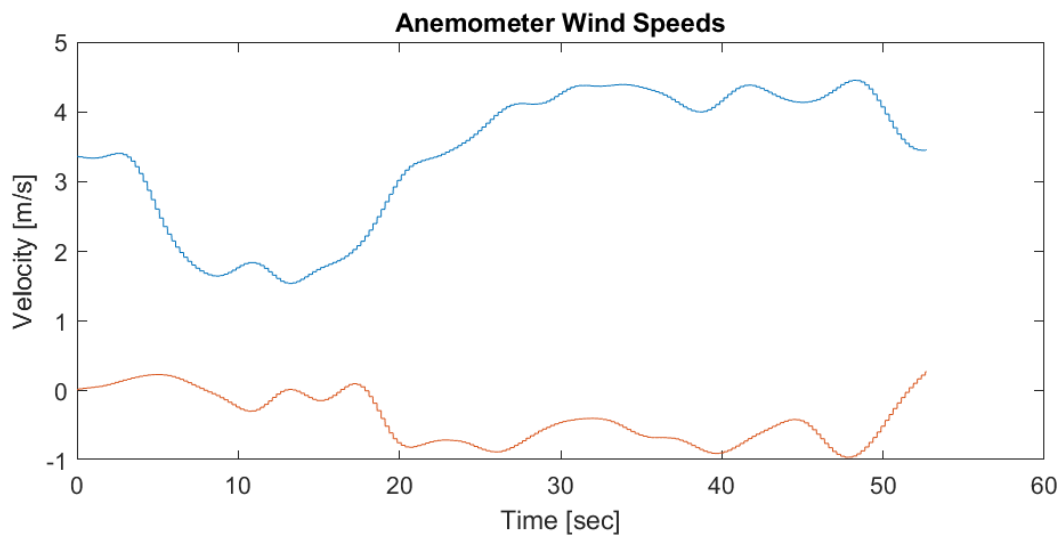
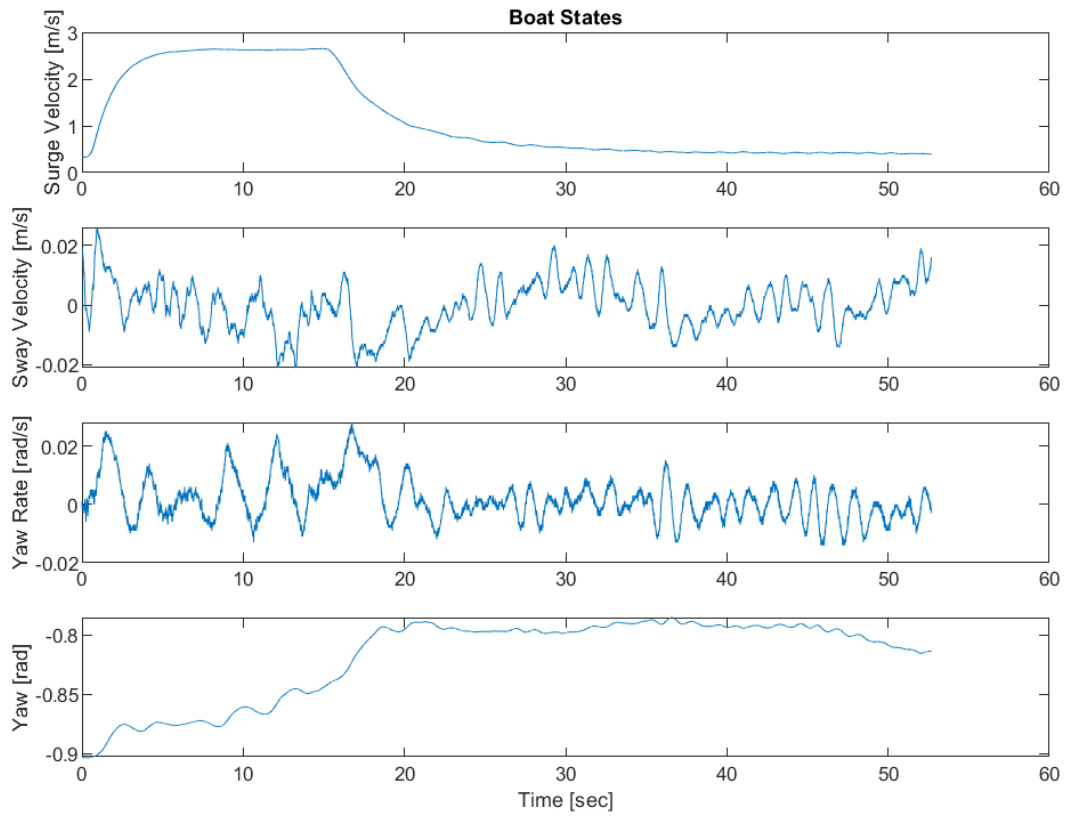


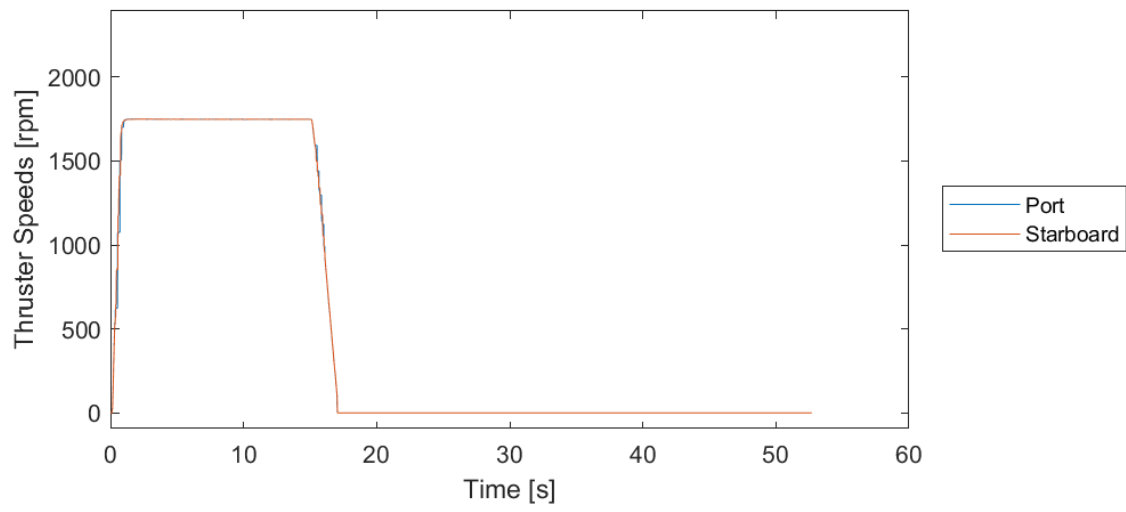
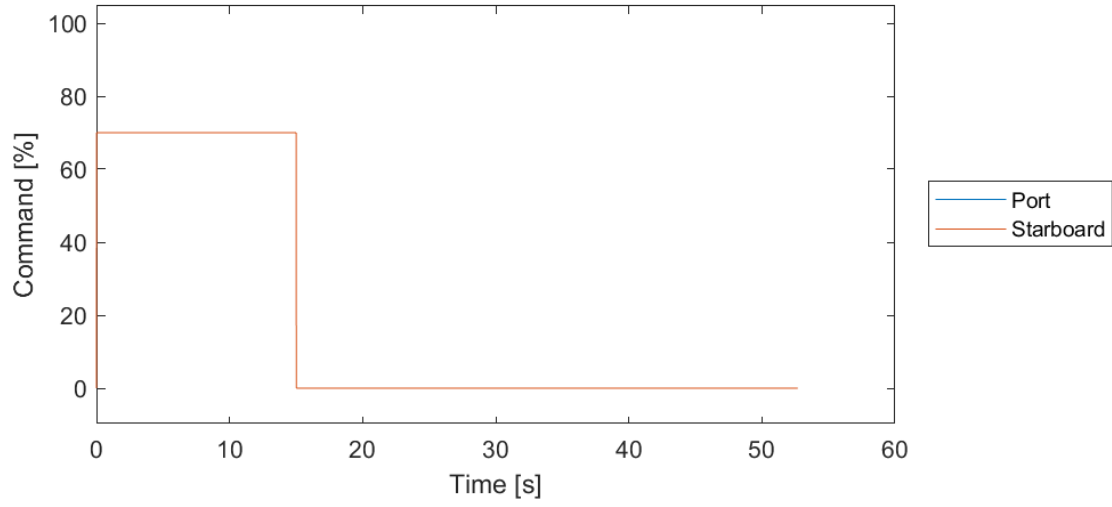
1.3.8. Stopping Set 8



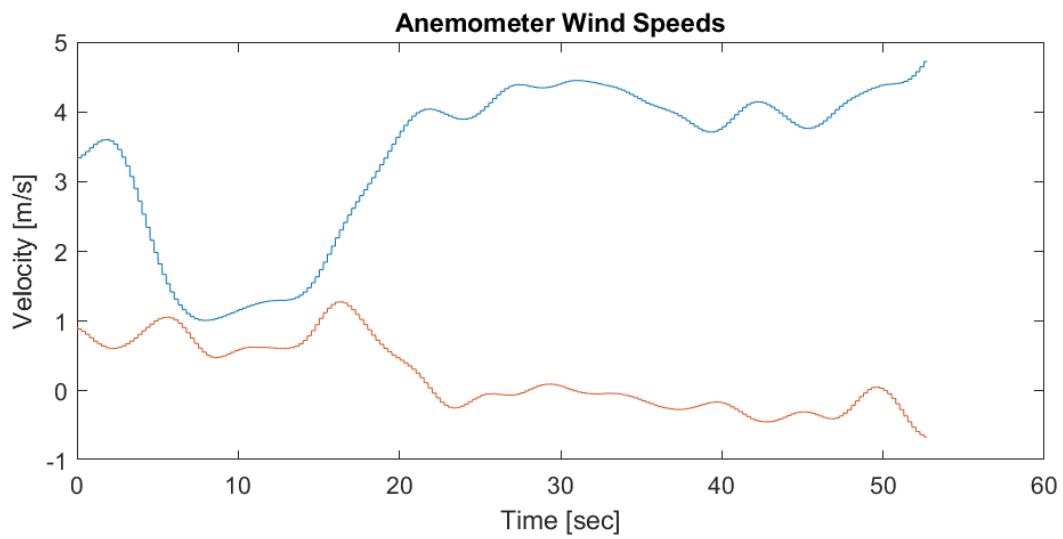
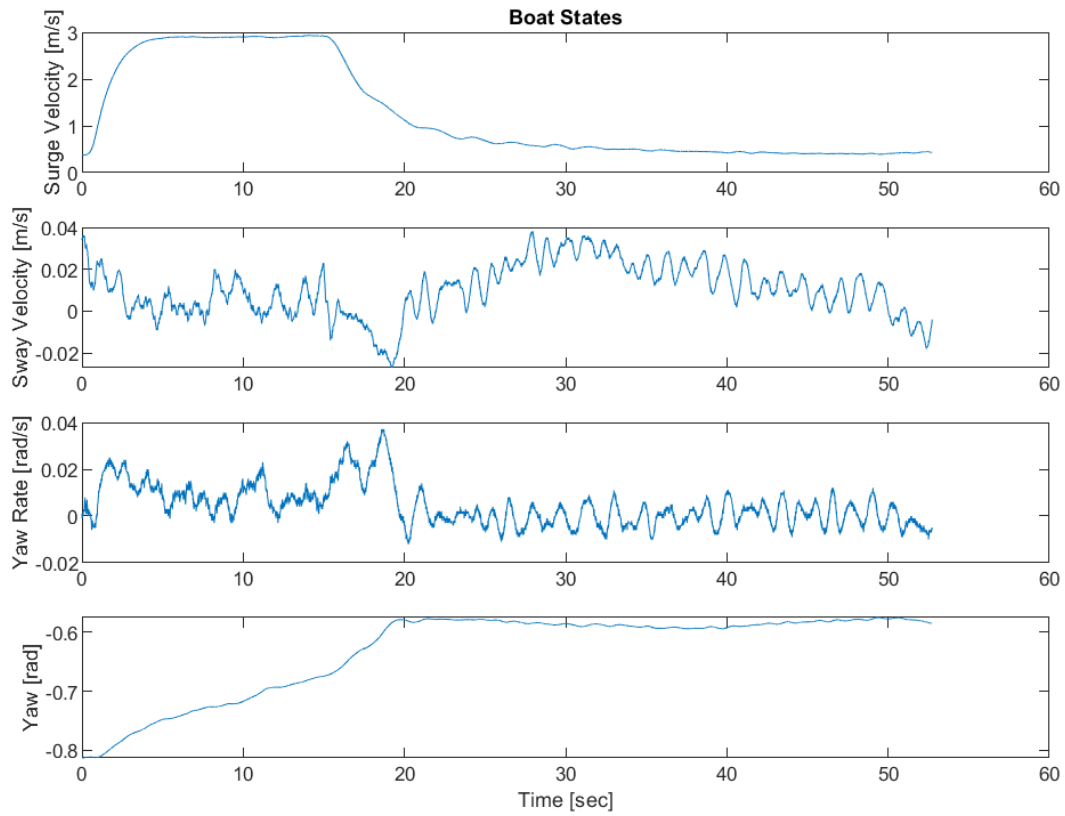


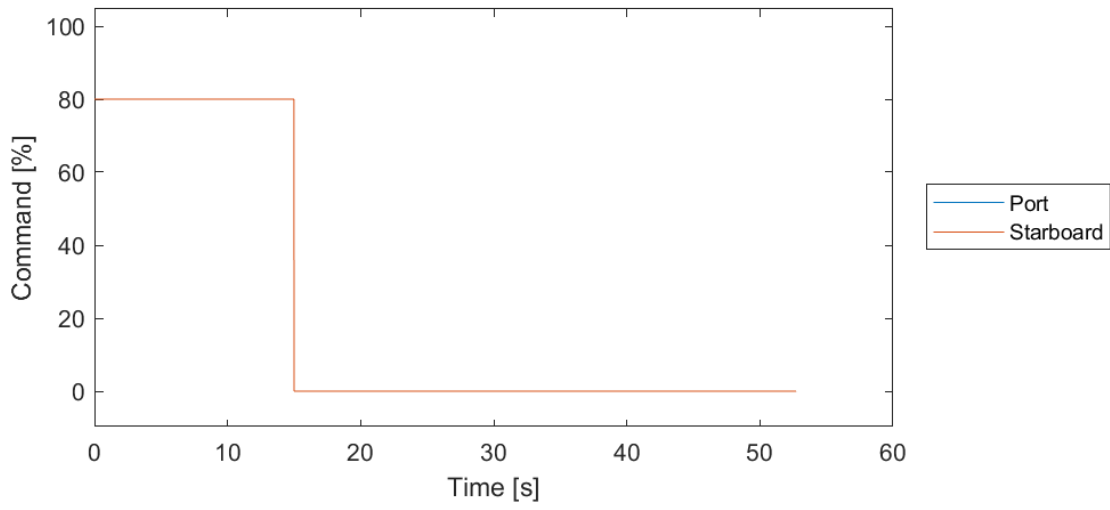
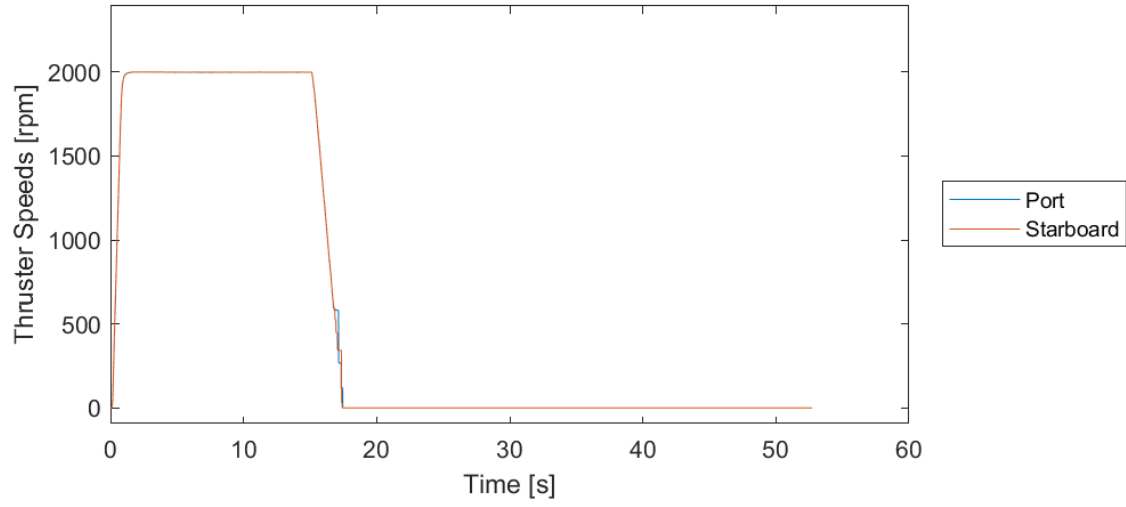
1.3.9. Stopping Set 9



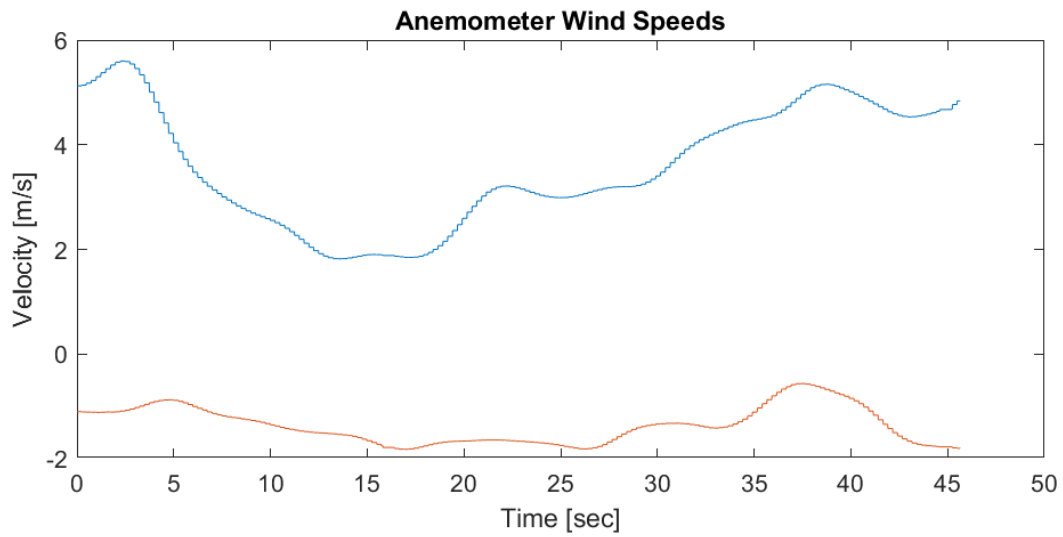
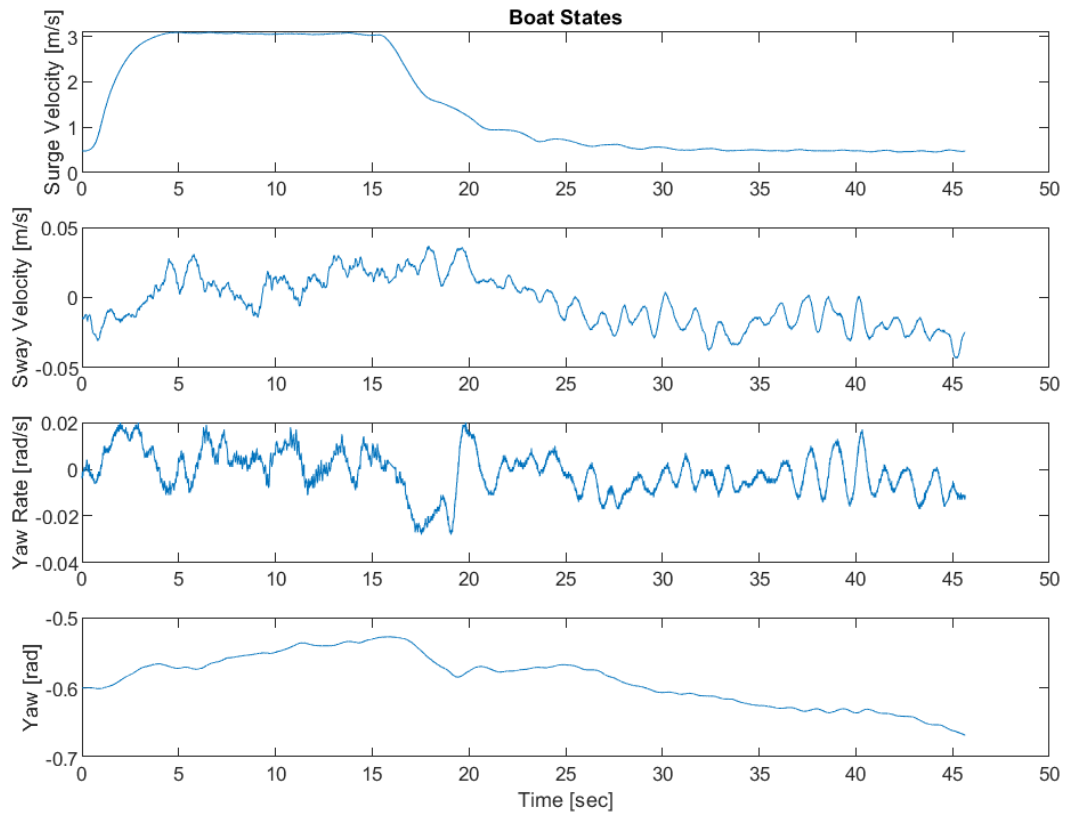


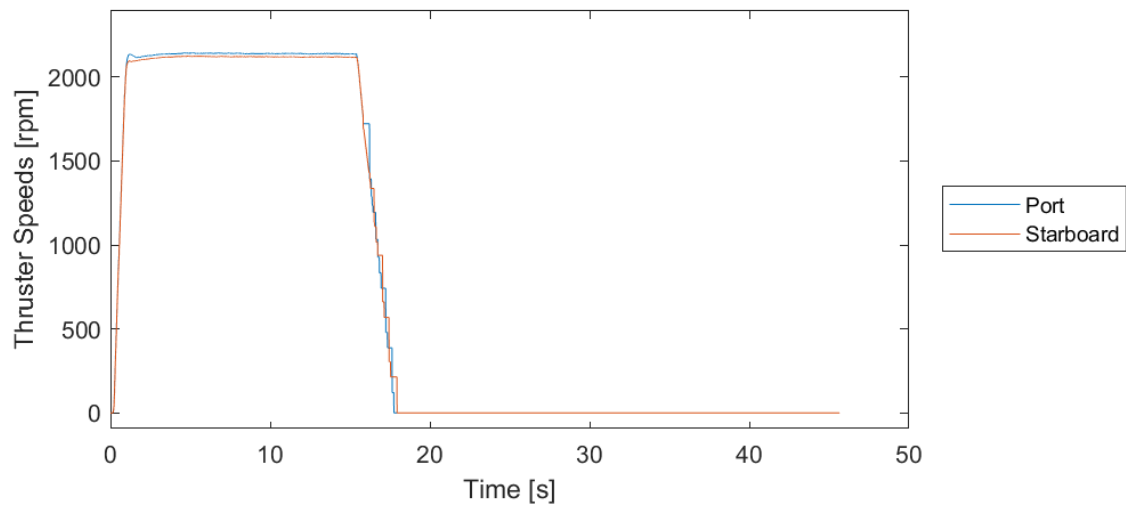
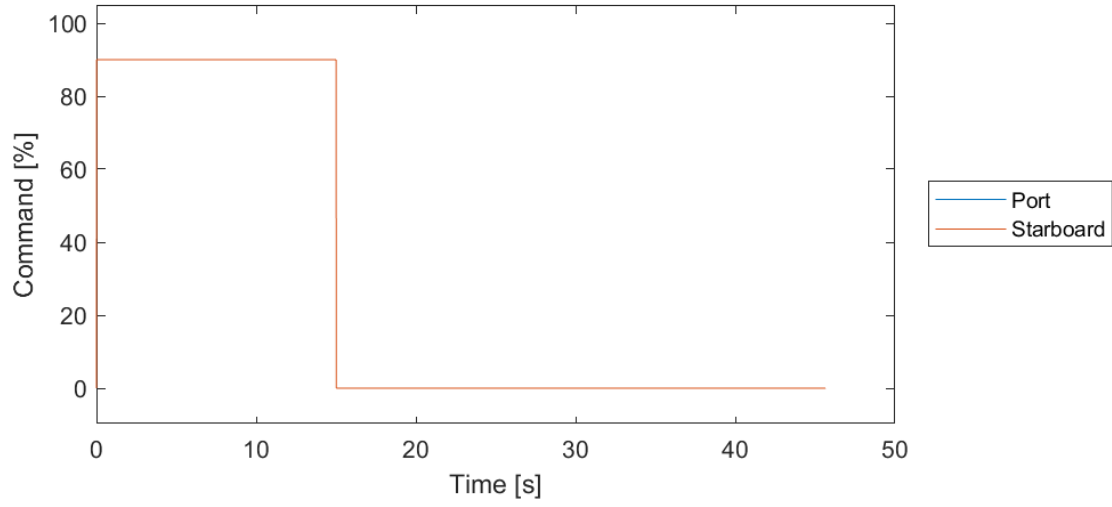
1.3.10. Stopping Set 10





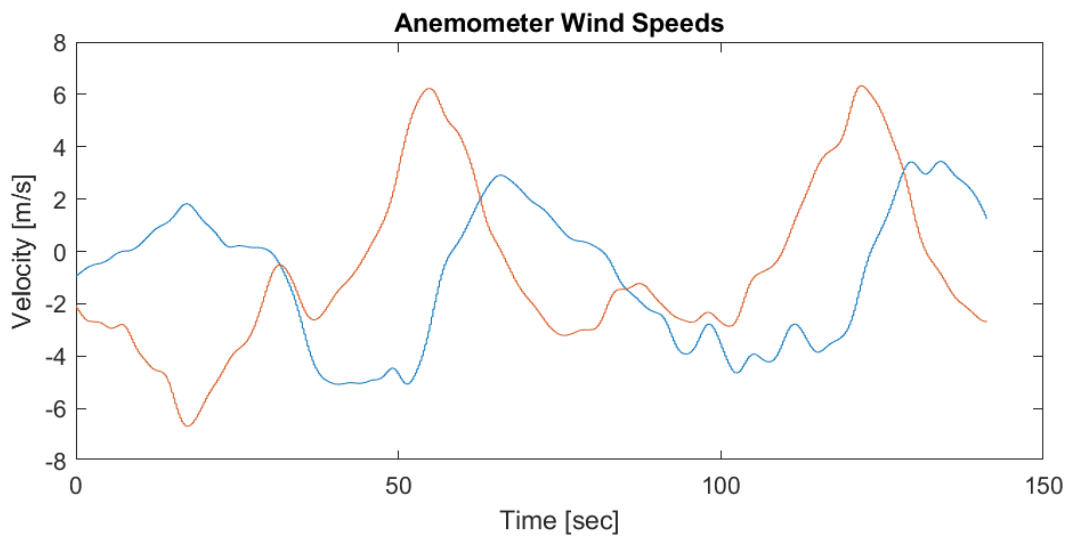
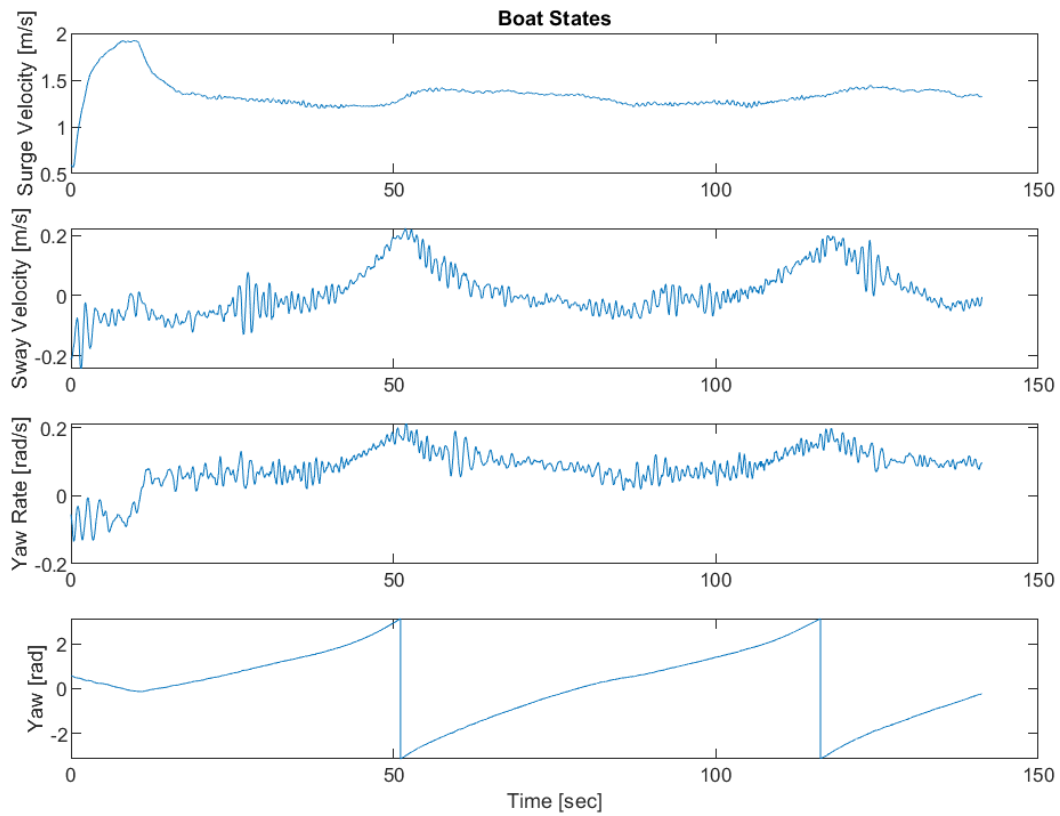
1.3.11. Stopping Set 11

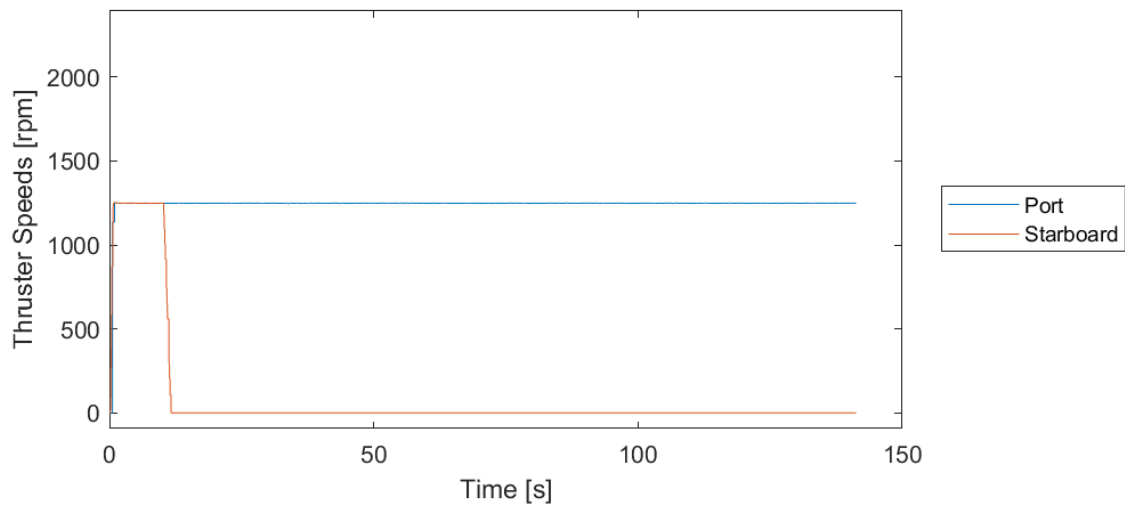
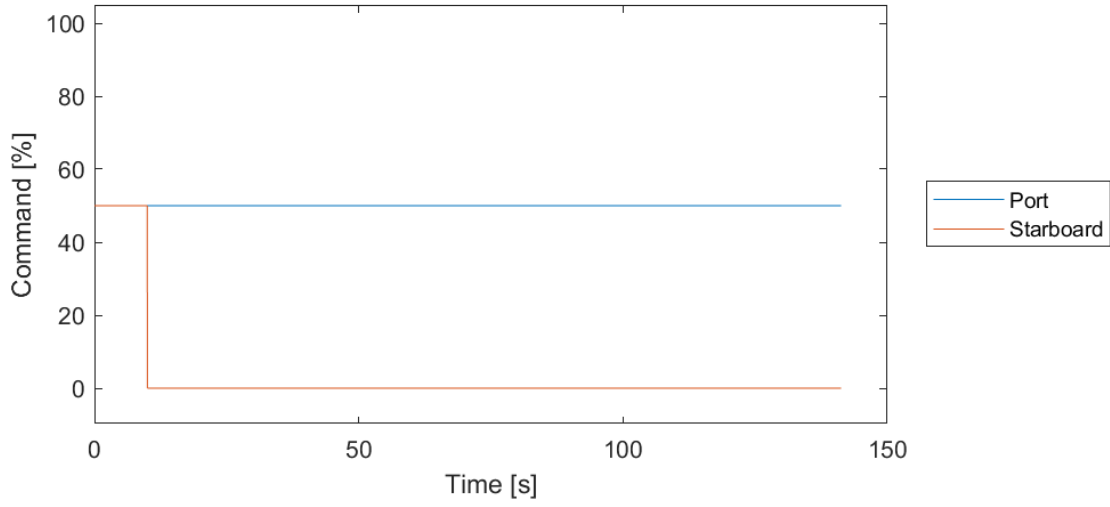




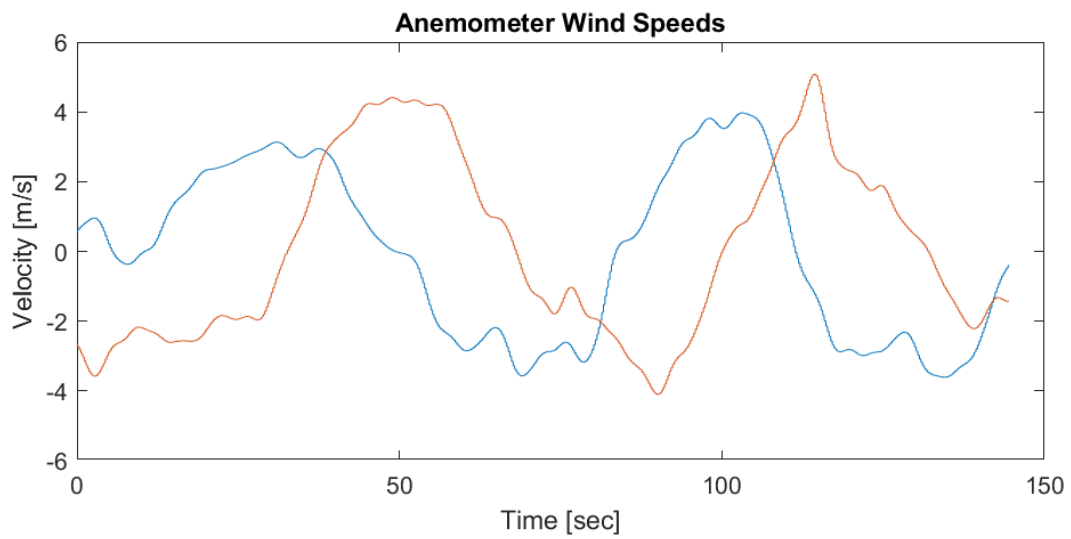
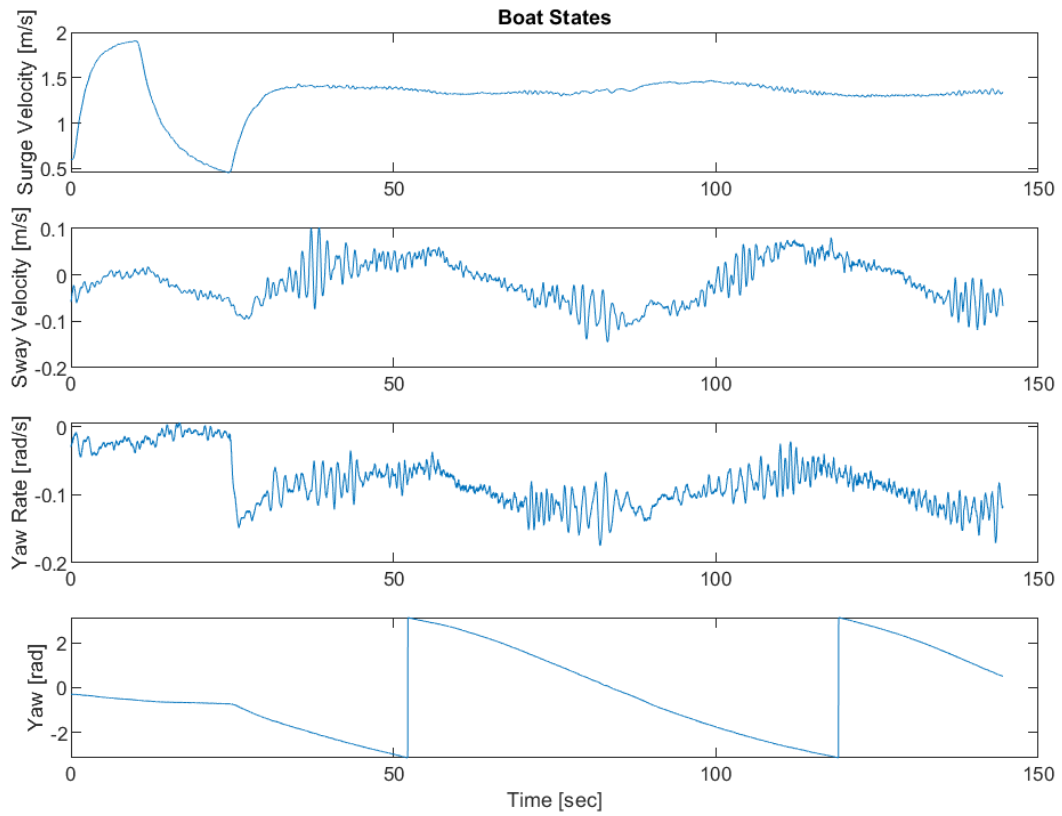
1.4. Turning Test Data Sets

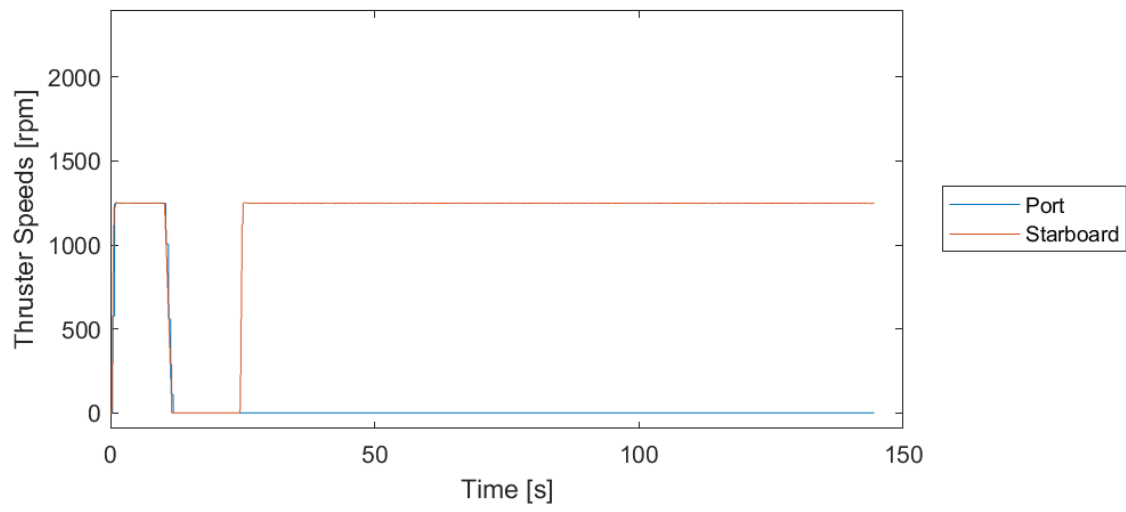
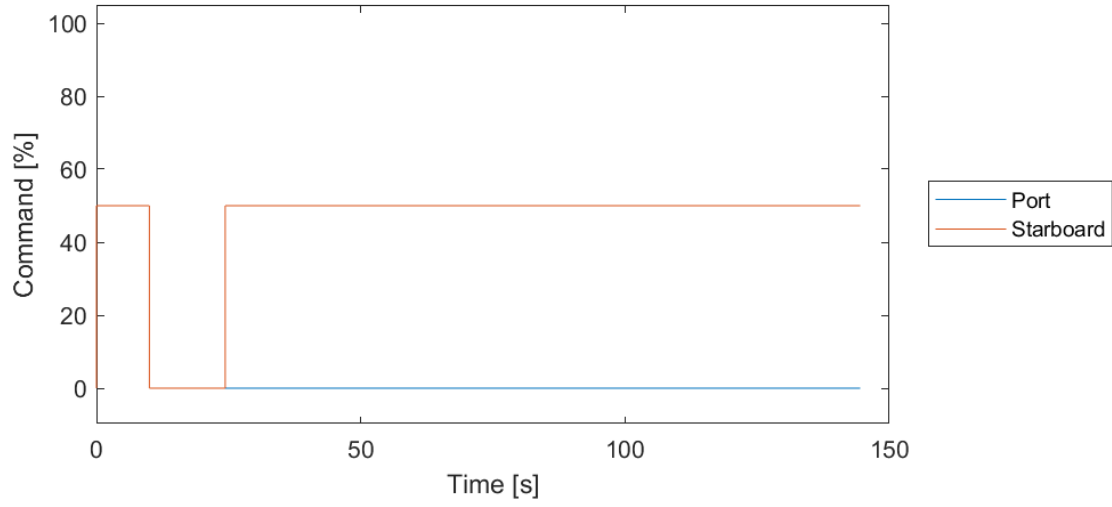
1.4.1. Turning Set 1



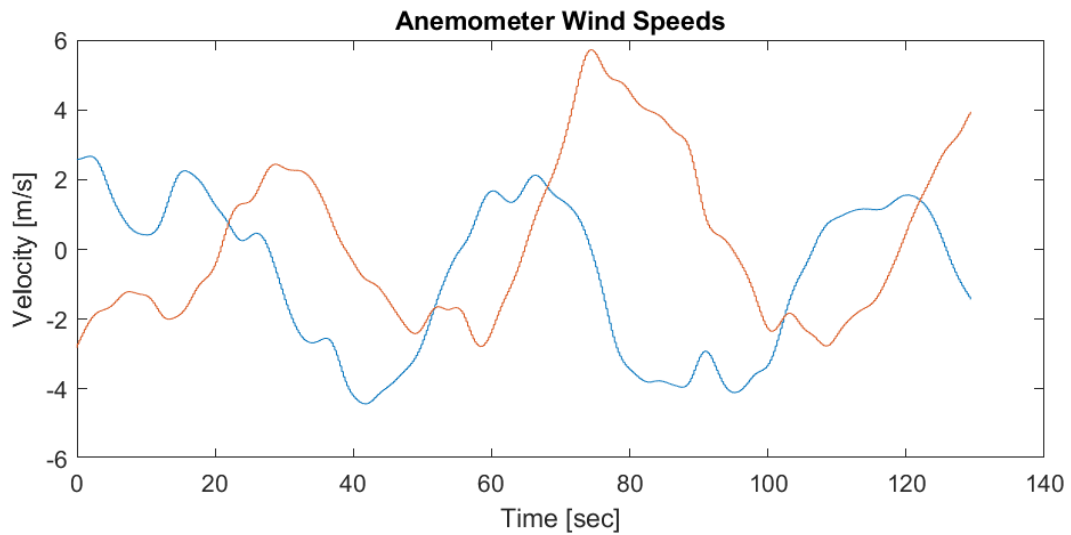
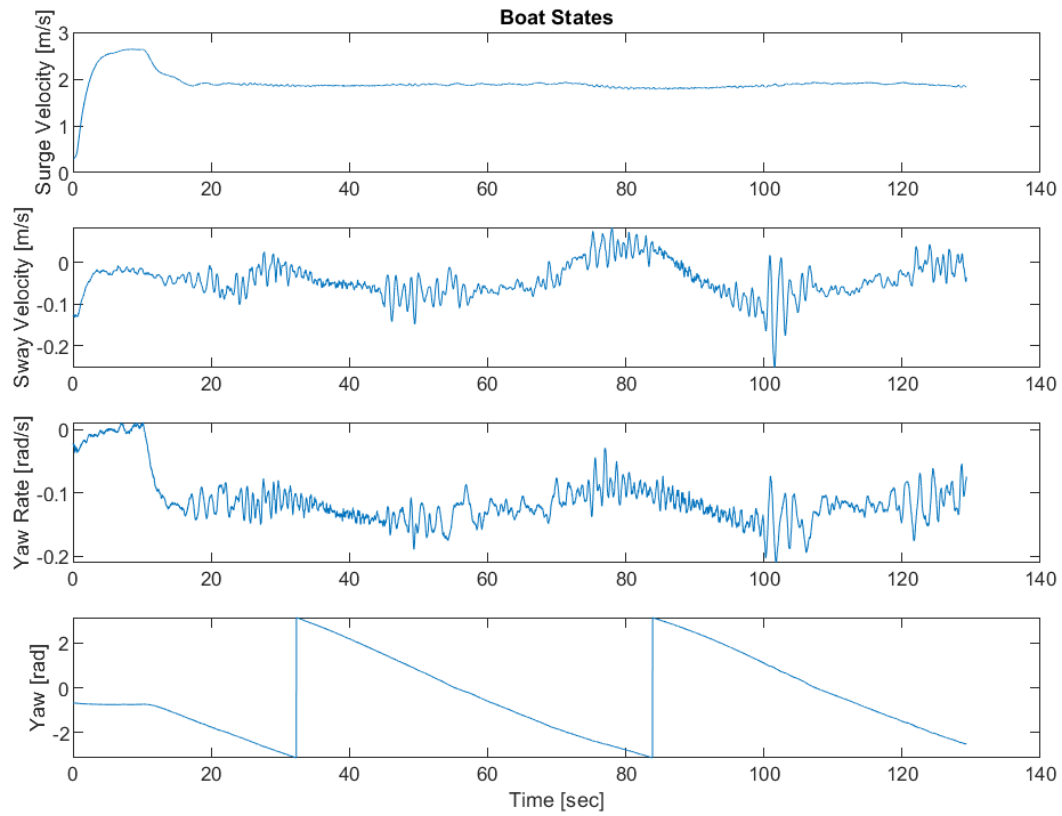


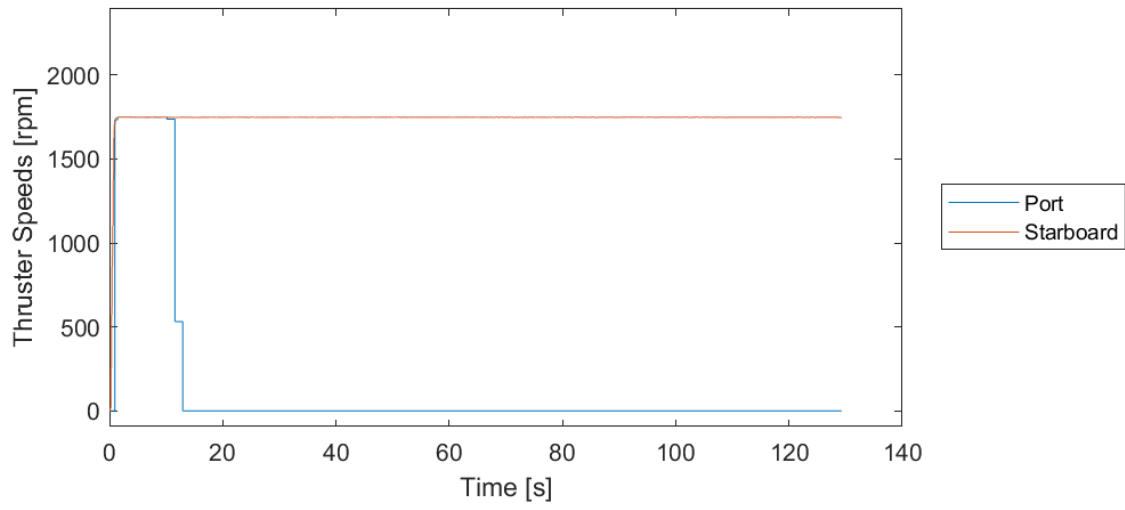
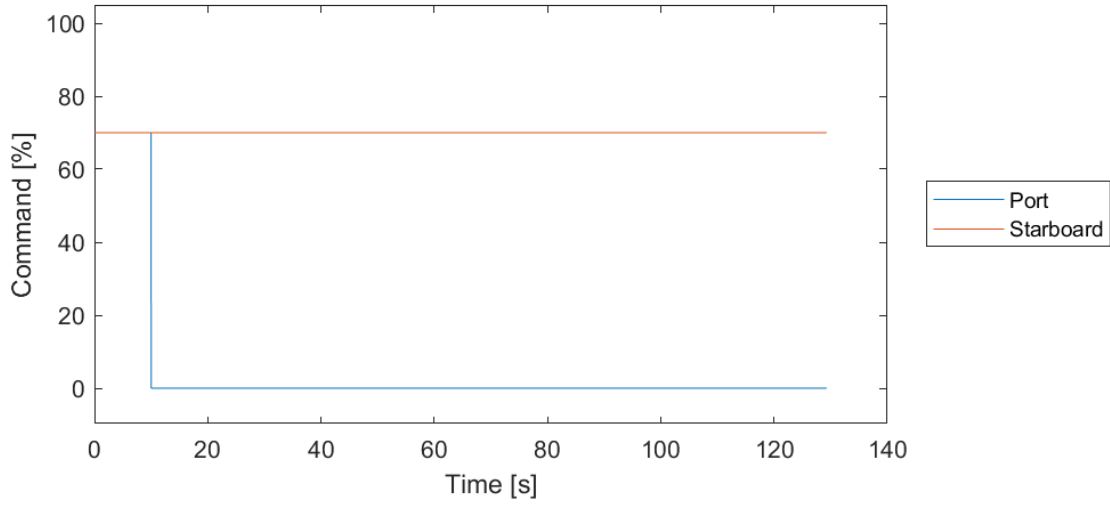
1.4.2. Turning Set 2



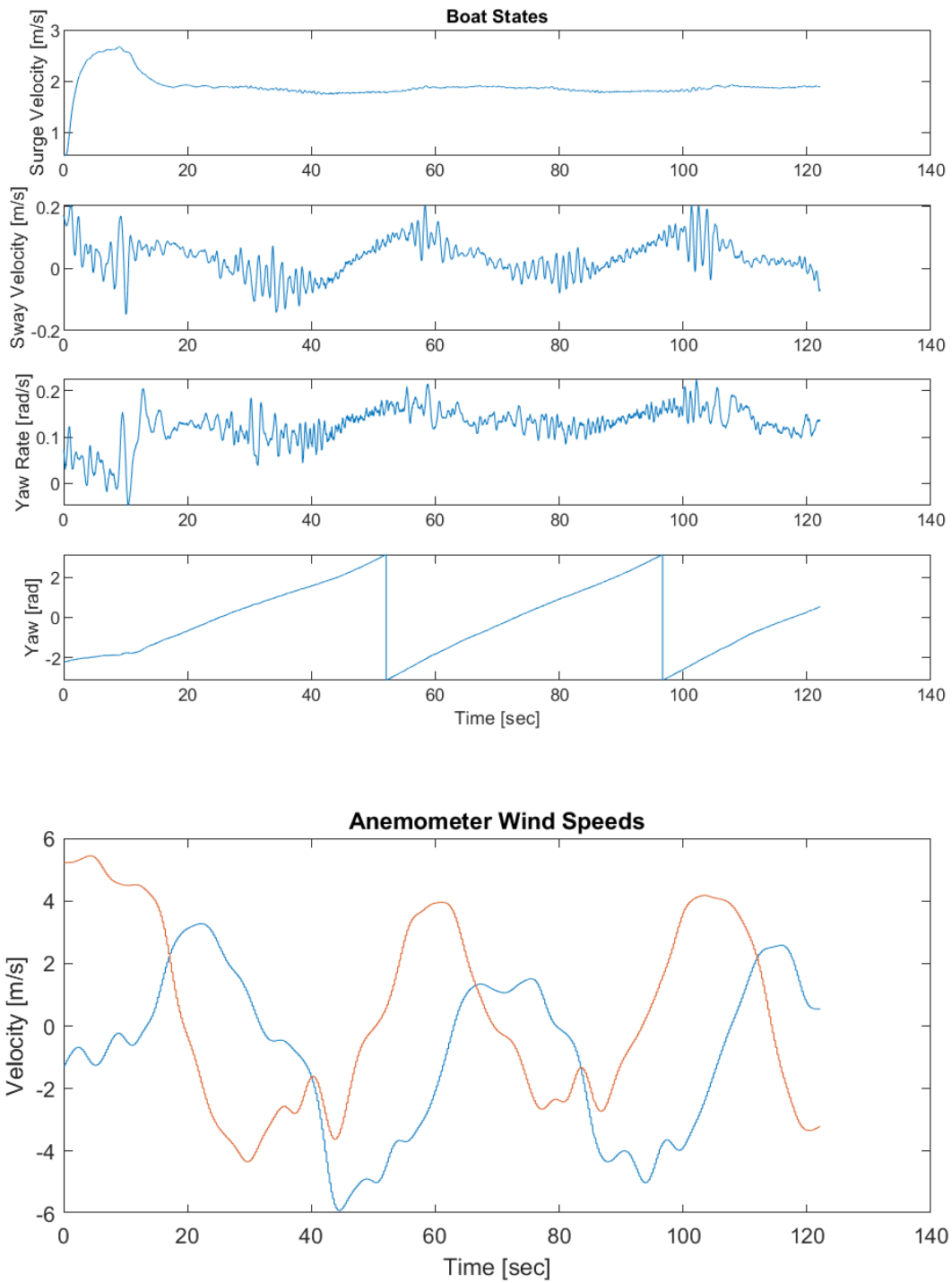


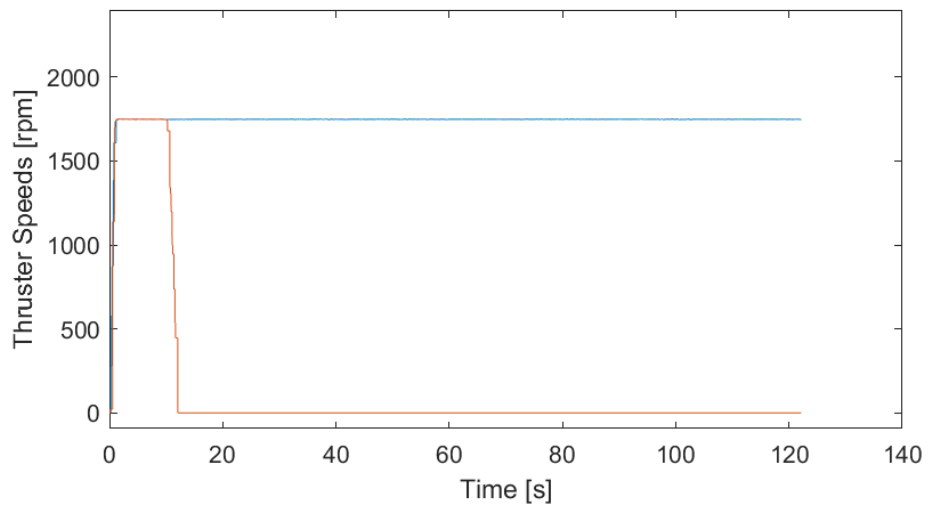
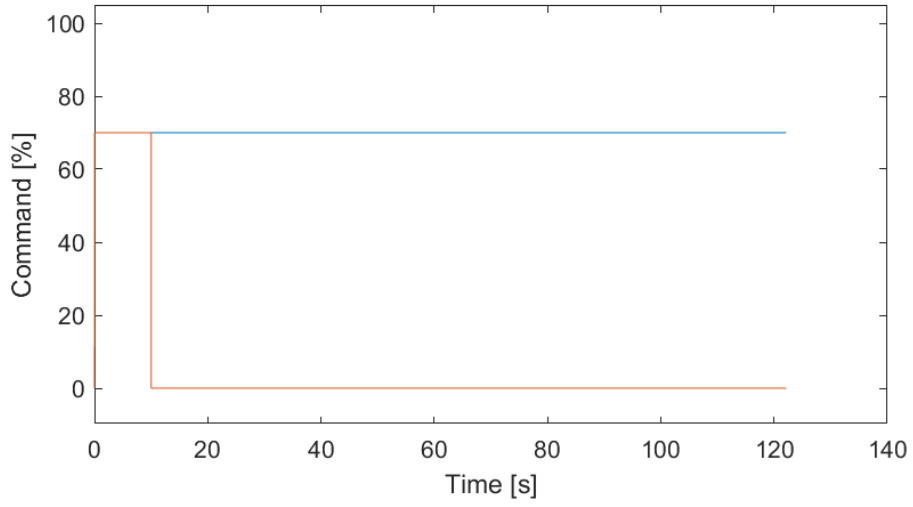
1.4.3. Turning Set 3



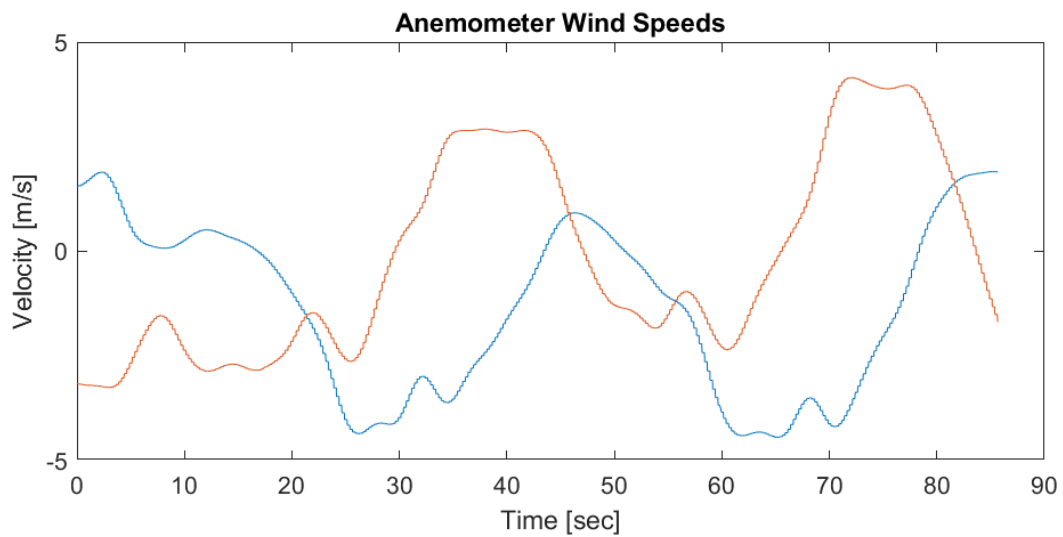
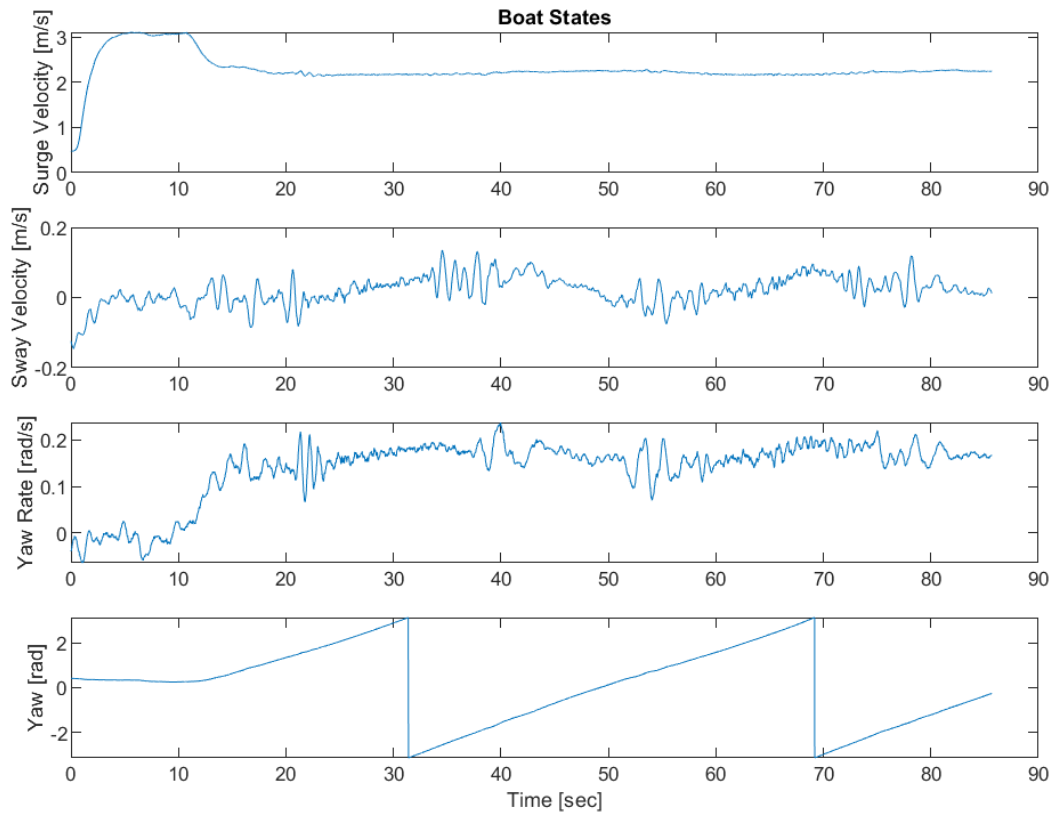


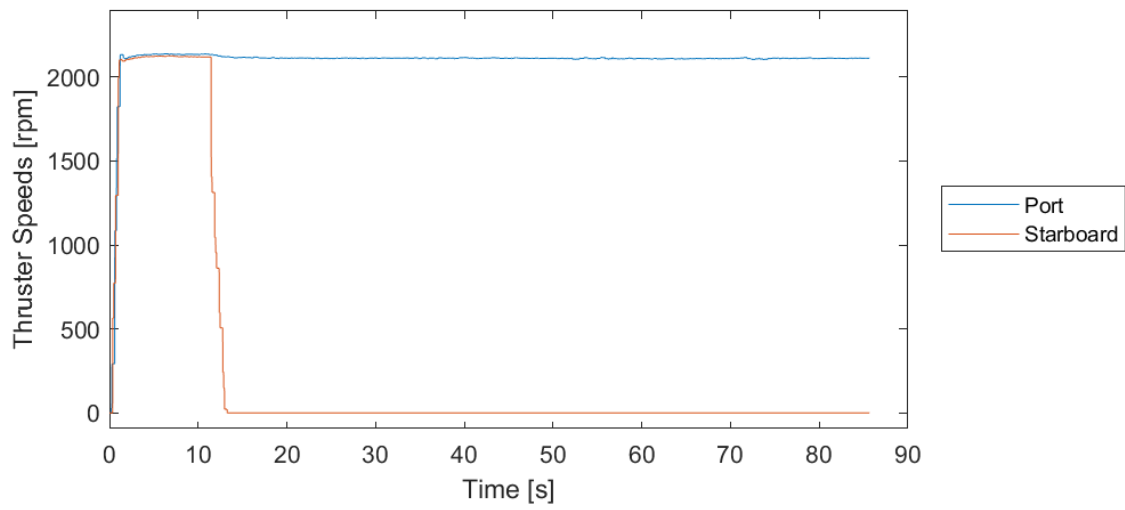
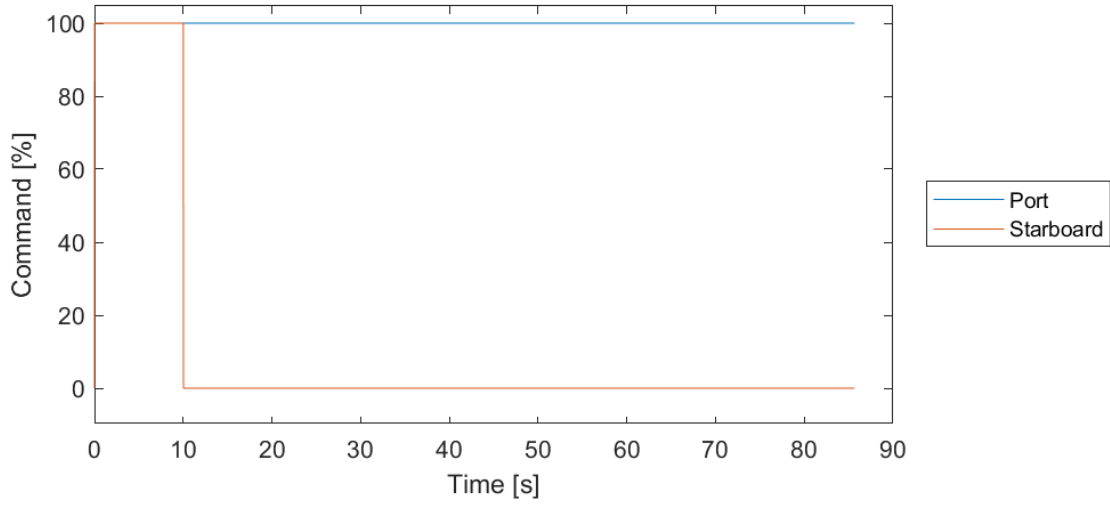
1.4.4. Turning Set 4



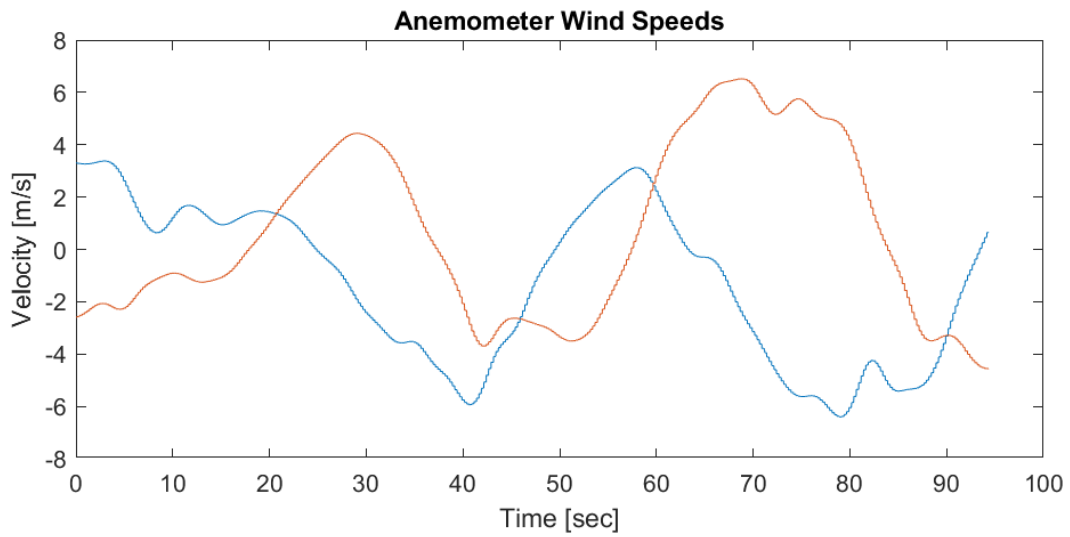
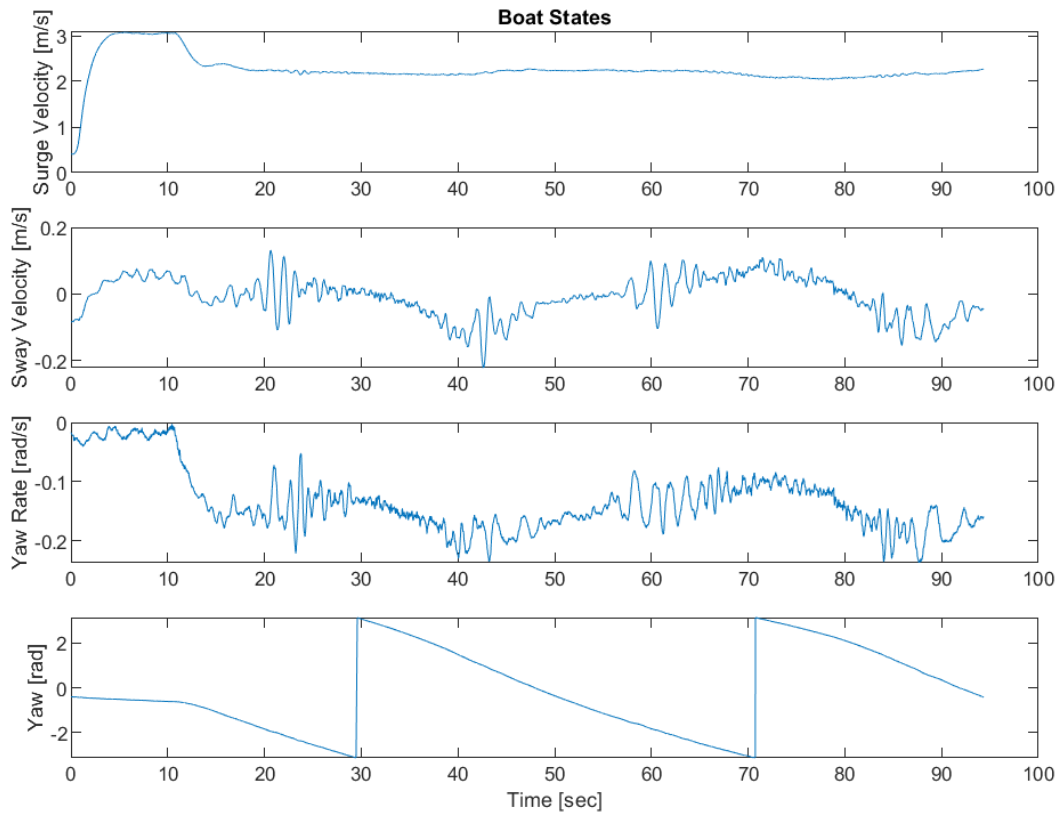


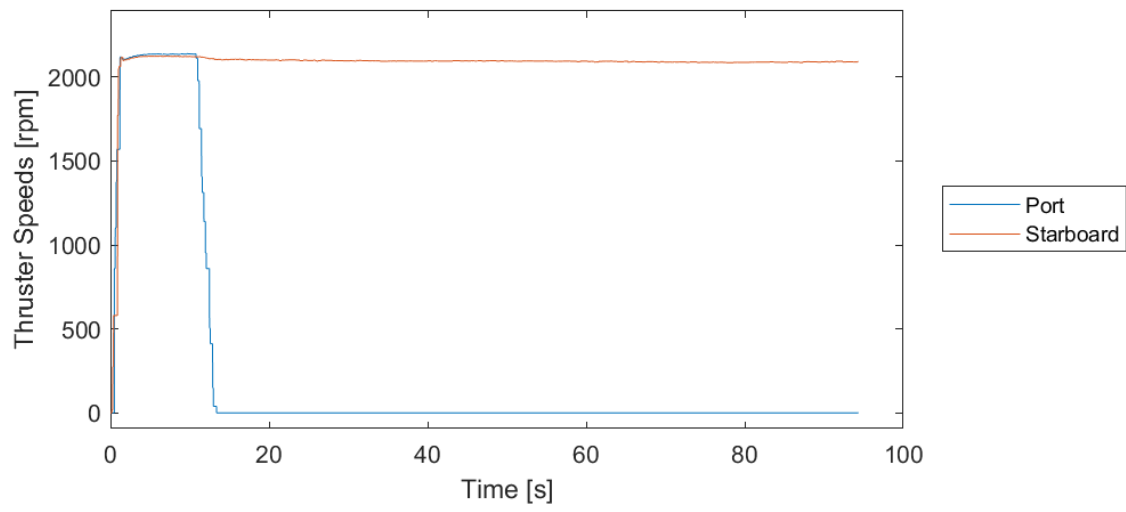
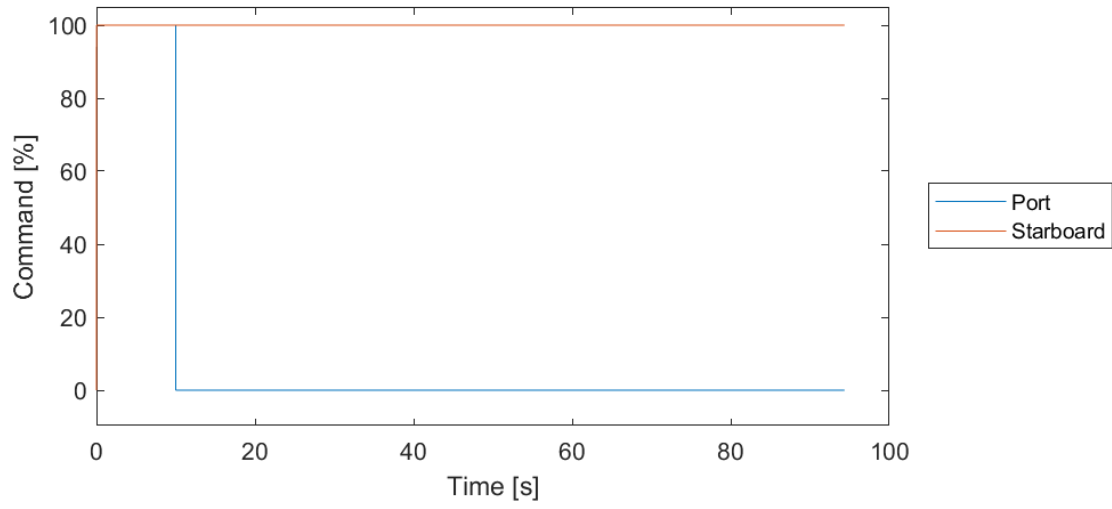
1.4.5. Turning Set 5



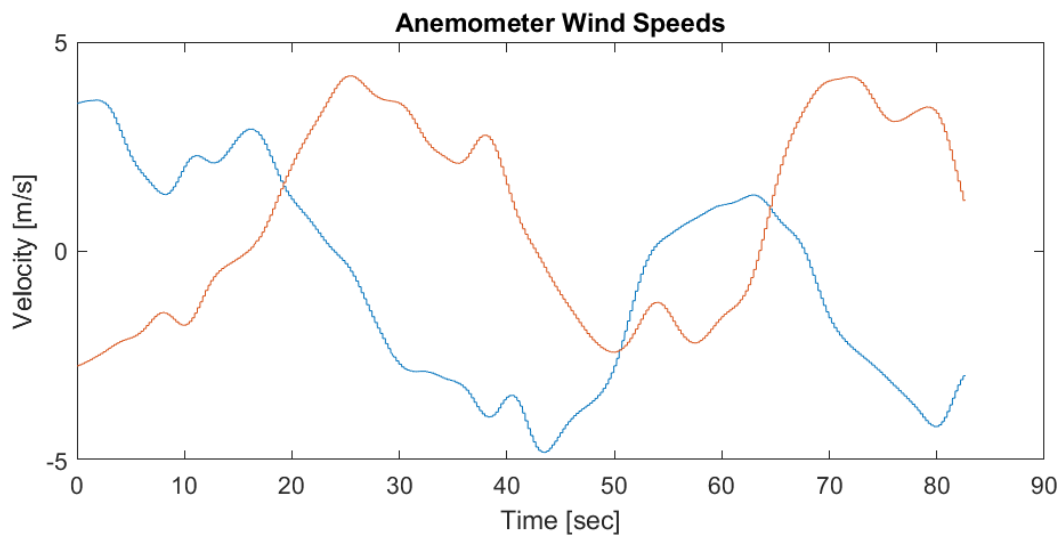
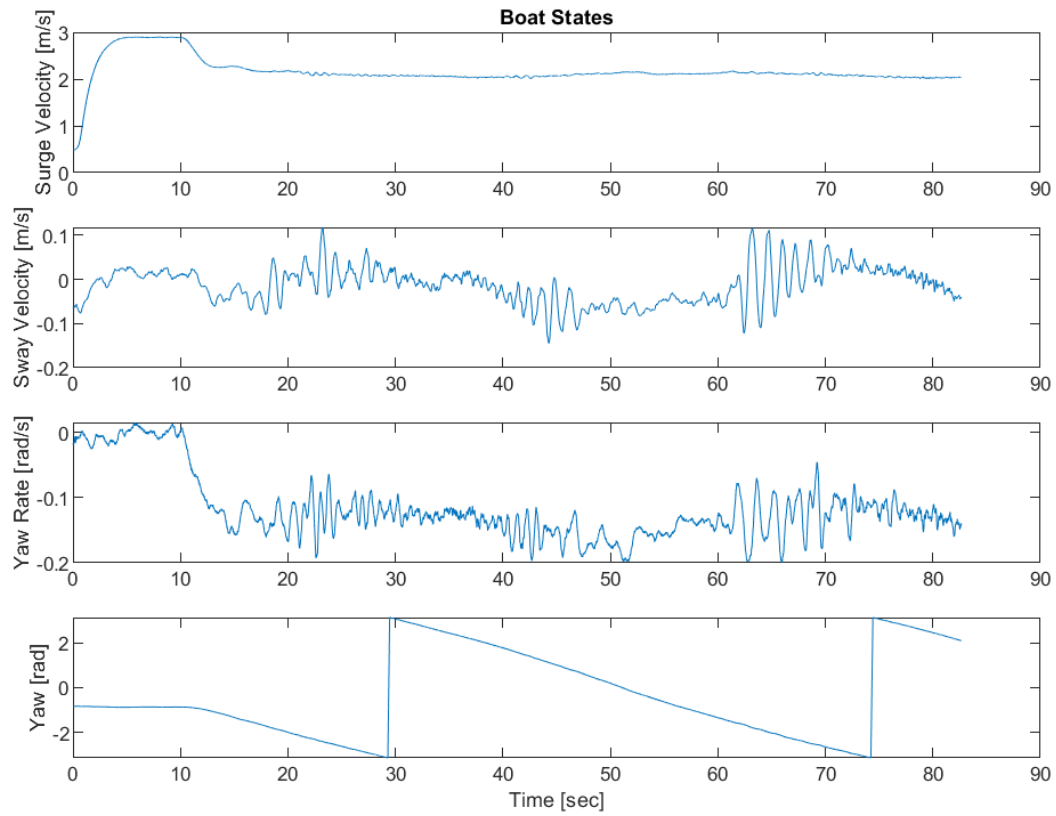


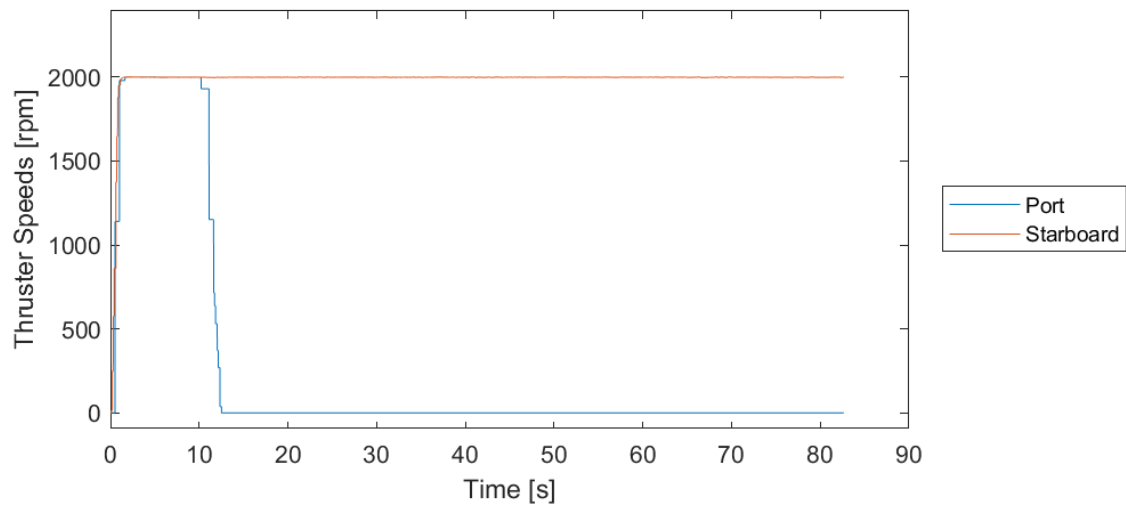
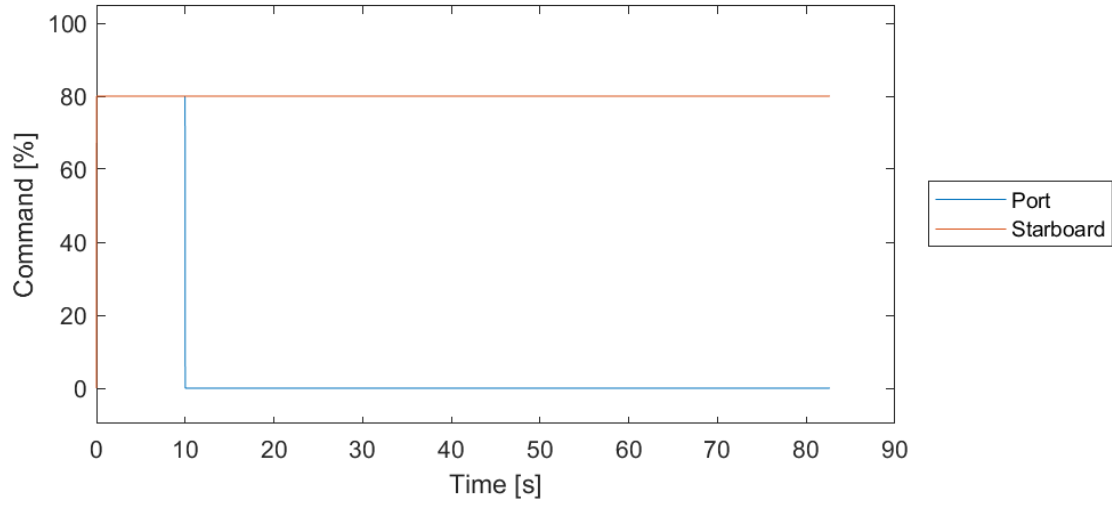
1.4.6. Turning Set 6





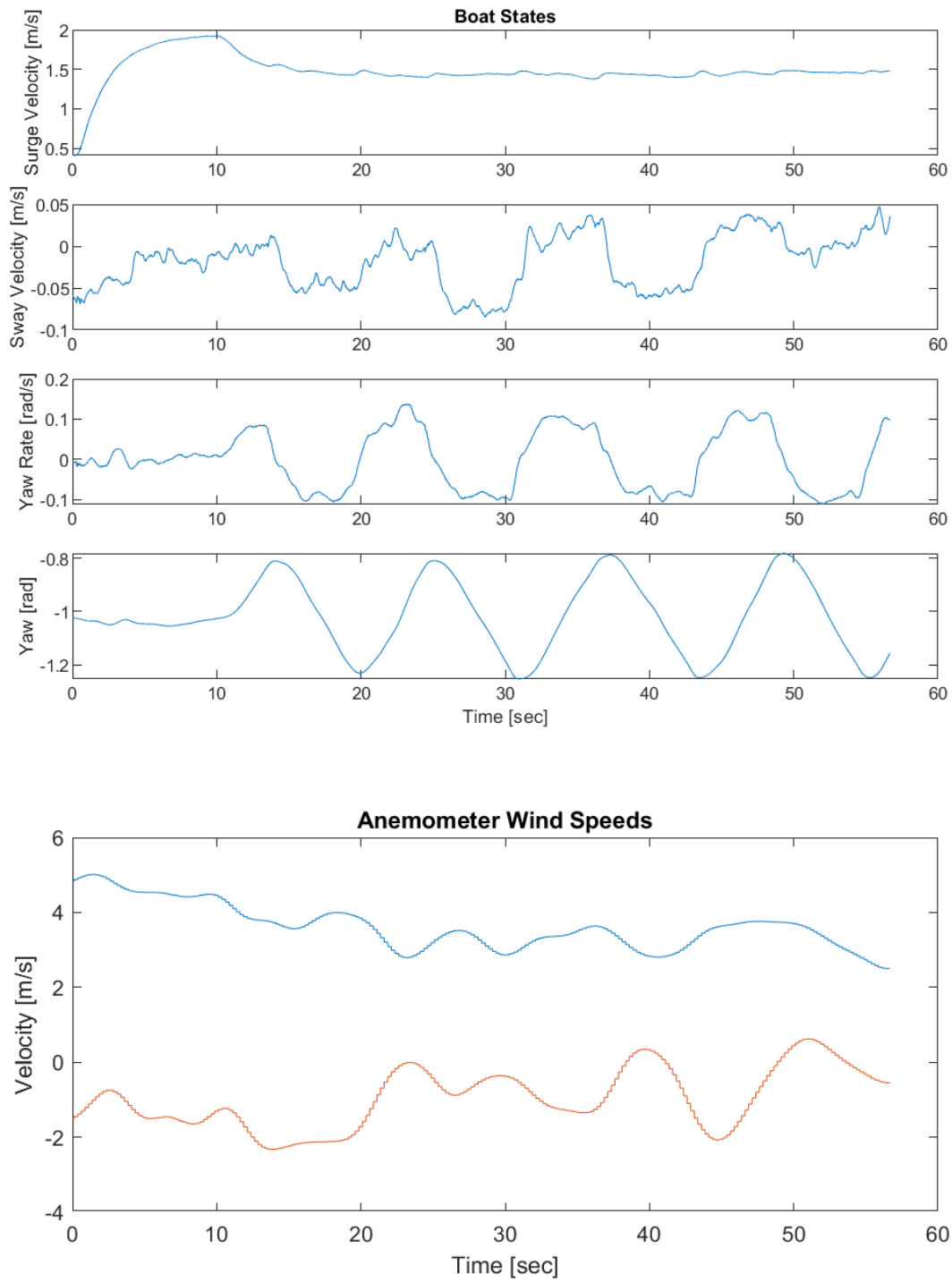
1.4.7. Turning Set 7

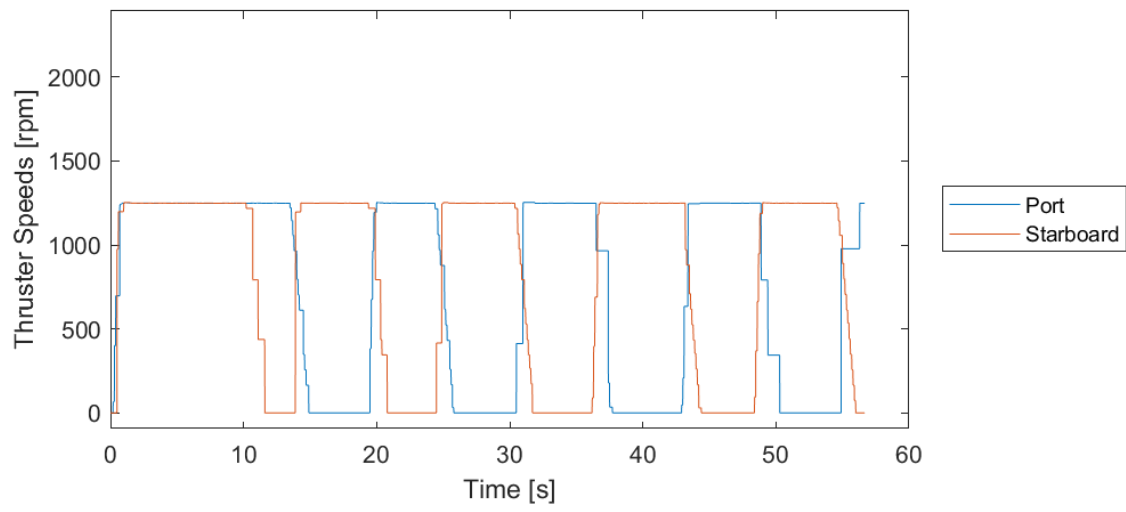
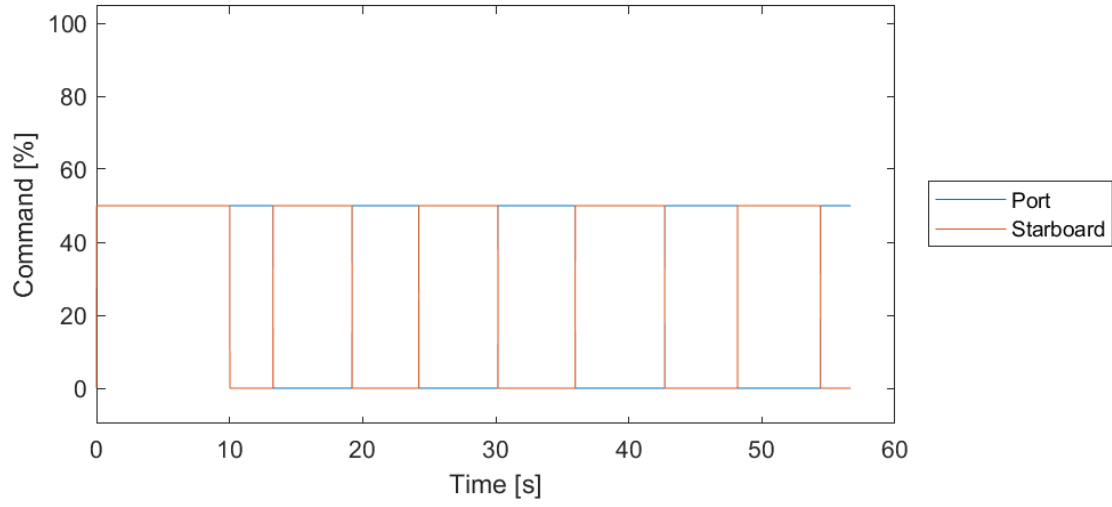




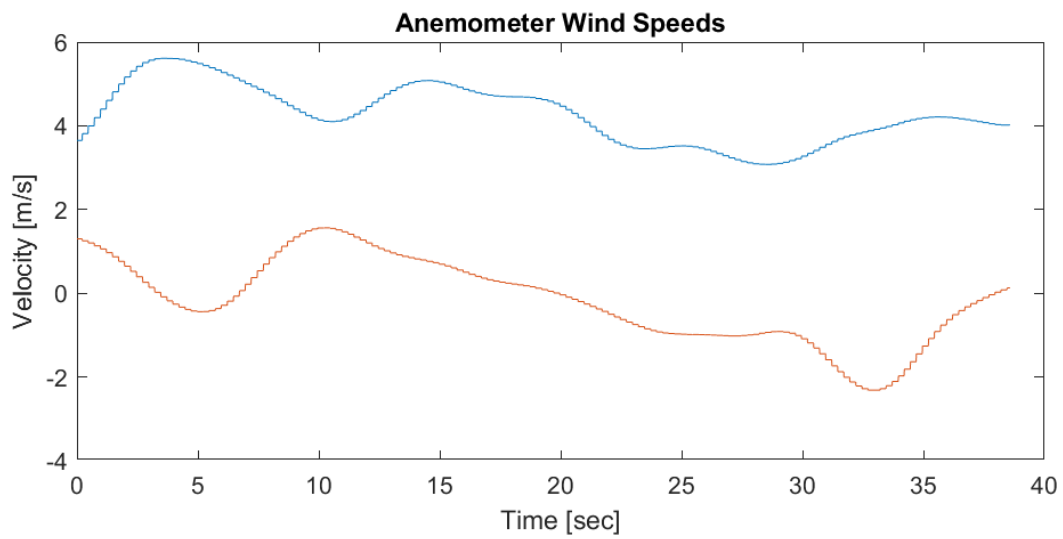
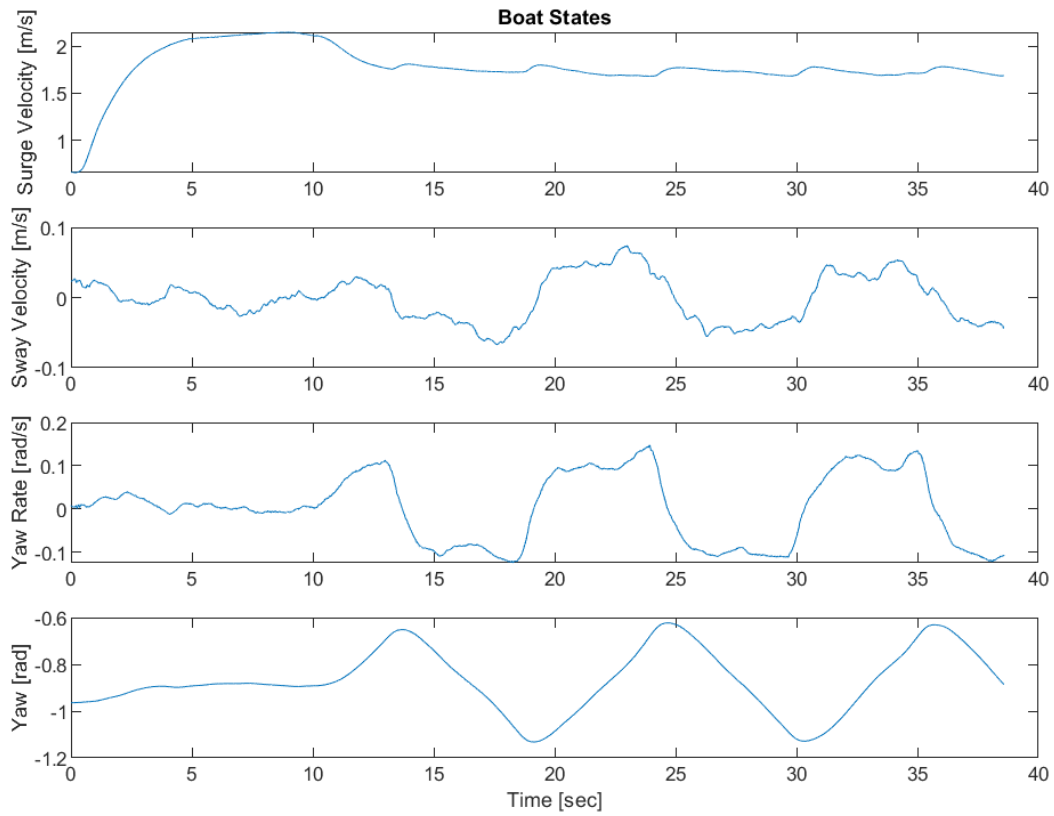
1.5. Zig-Zag Data Sets

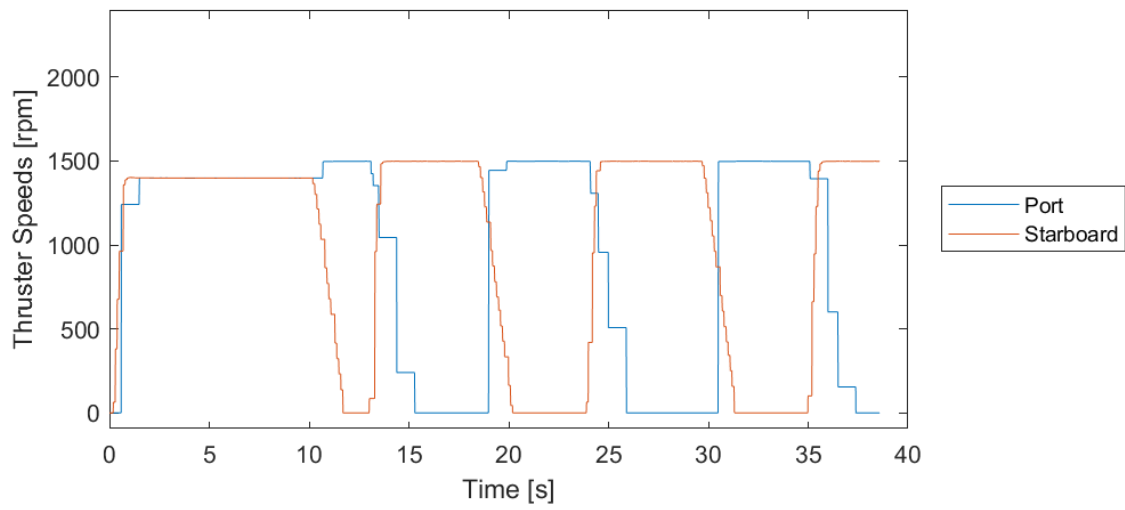
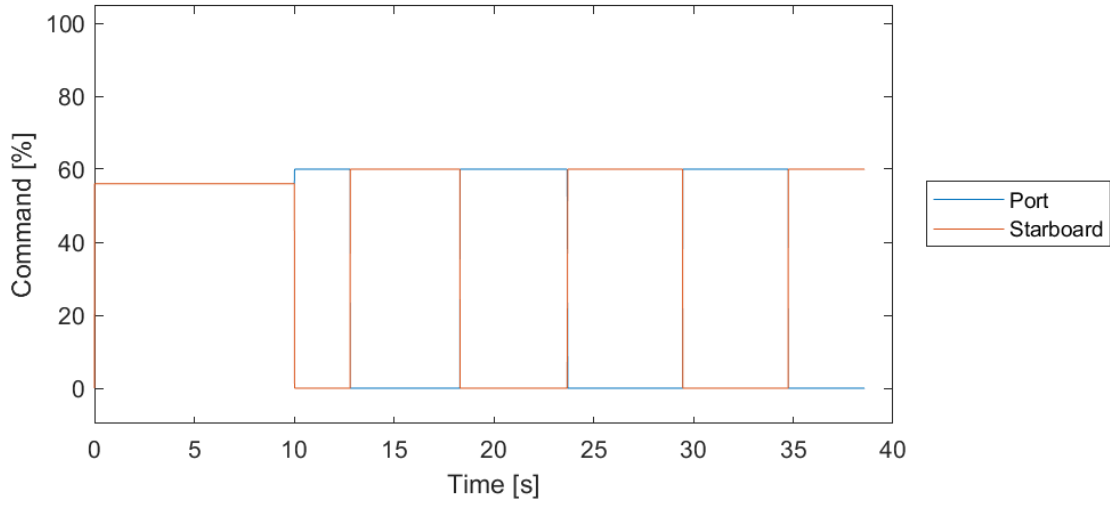
1.5.1. Zig-Zag Set 1



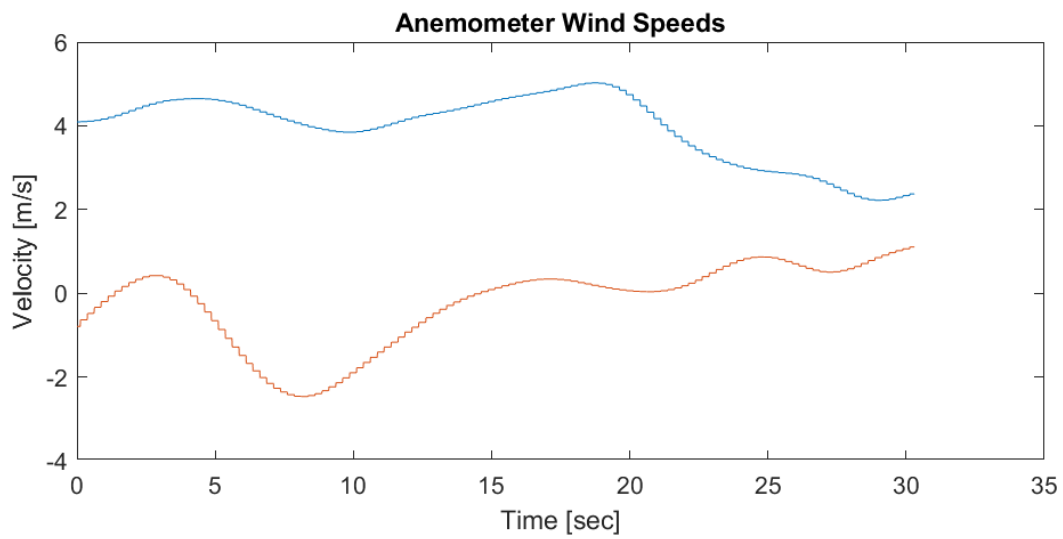
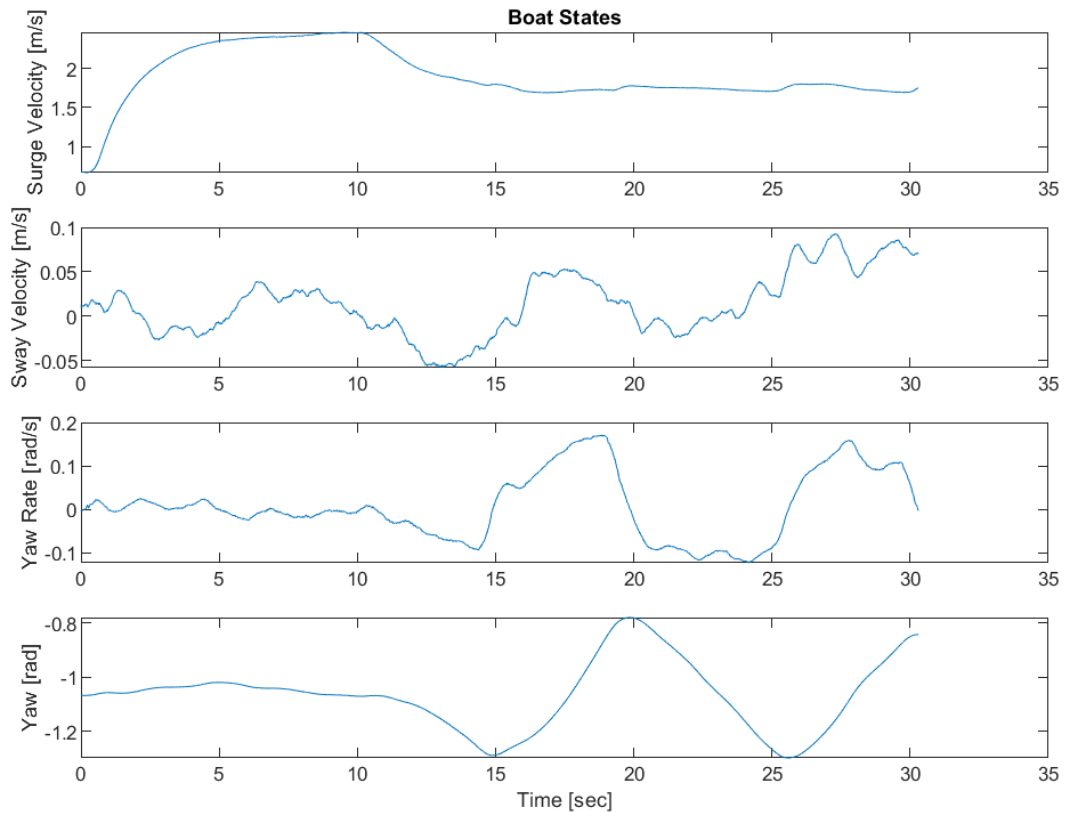


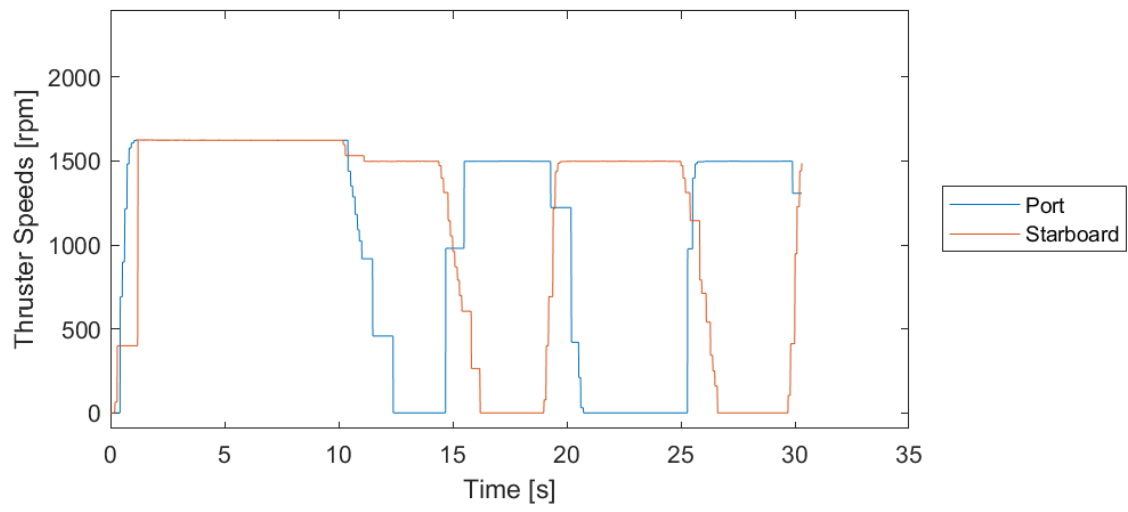
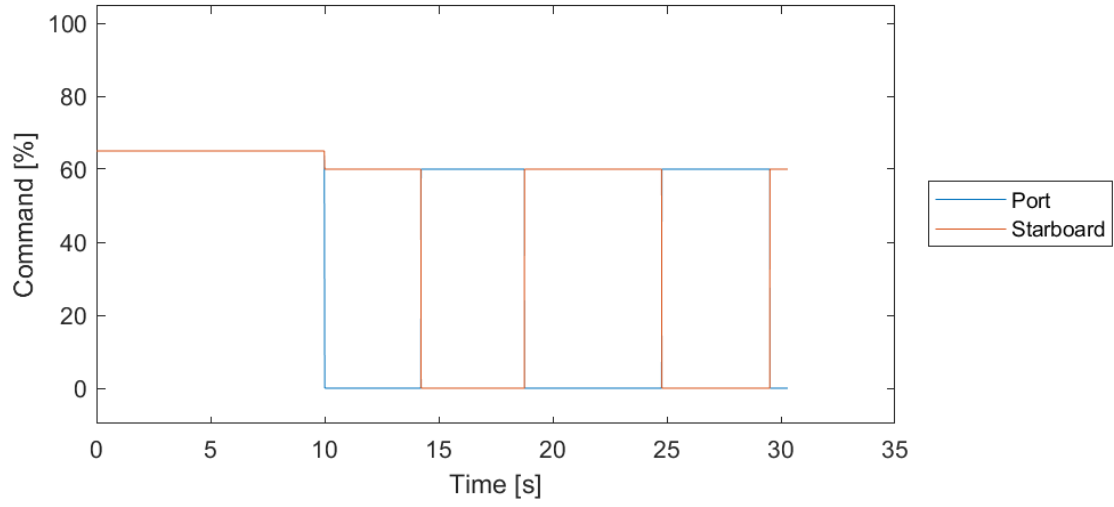
1.5.2. Zig-Zag Set 2



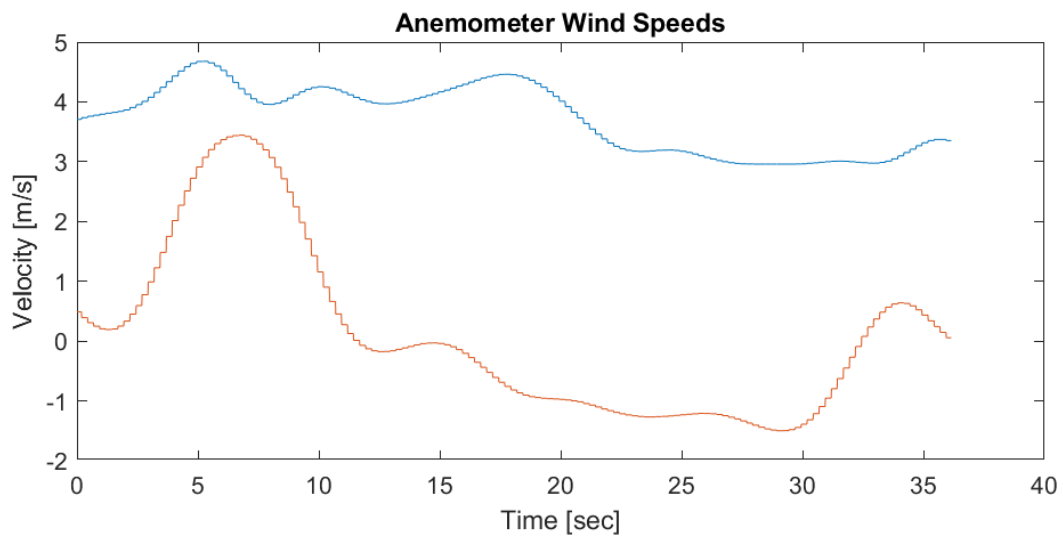
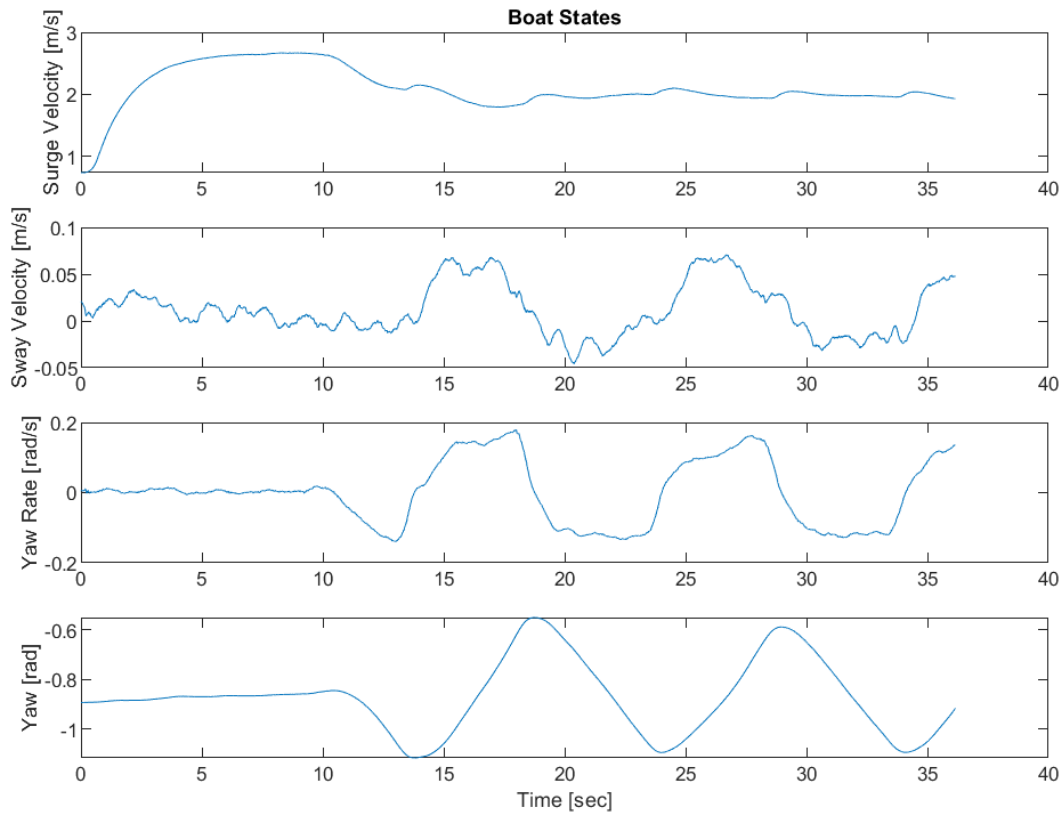


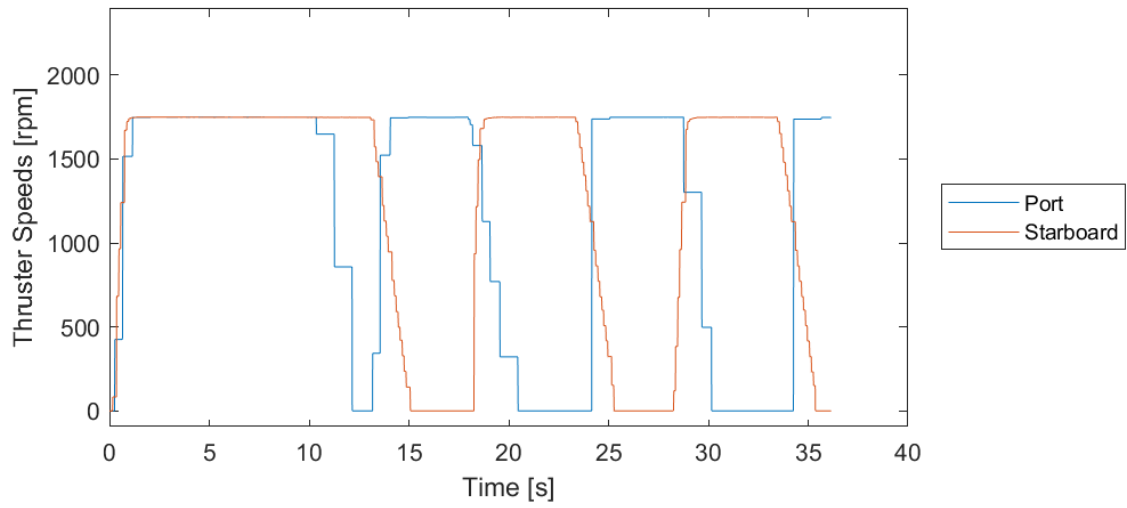
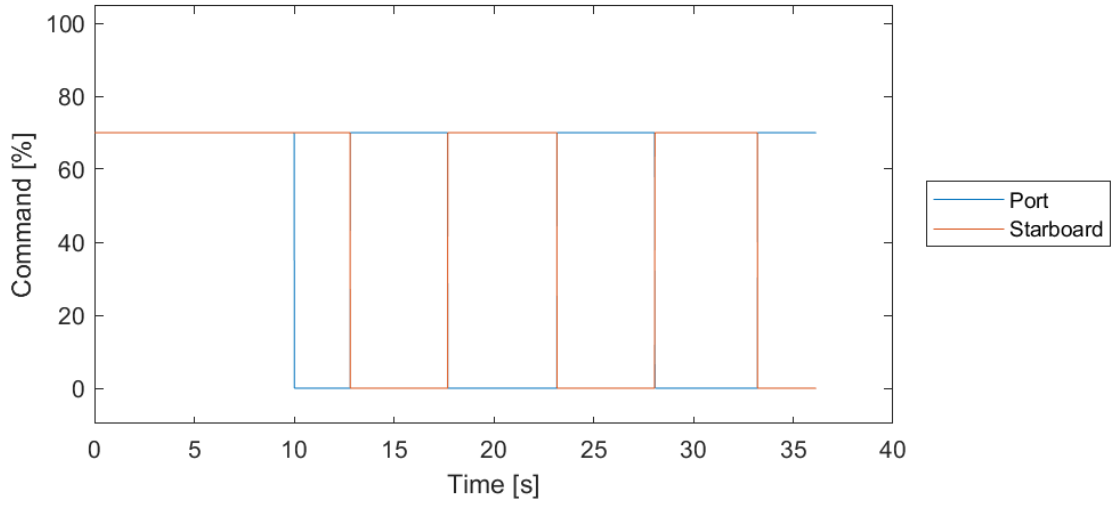
1.5.3. Zig-Zag Set 3



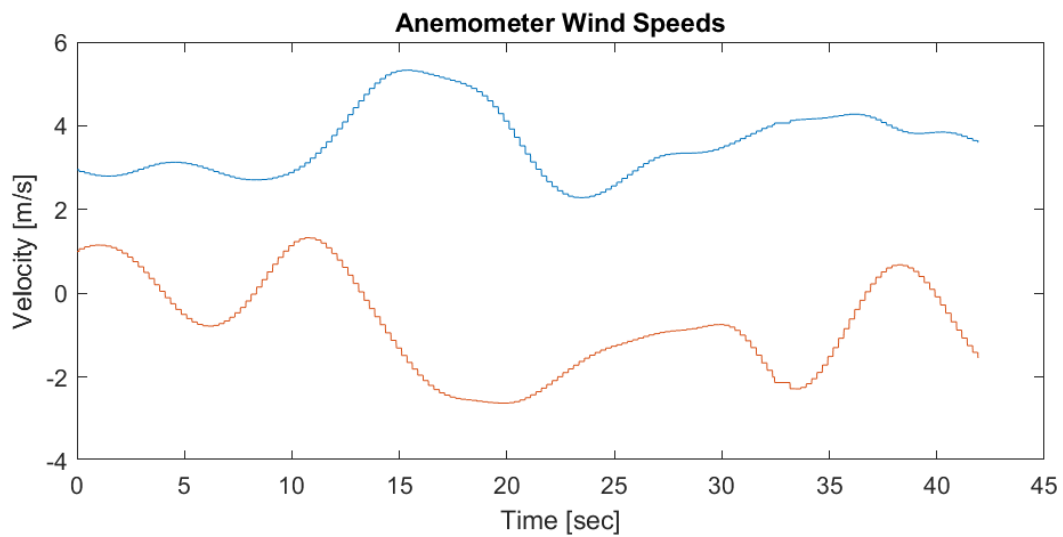
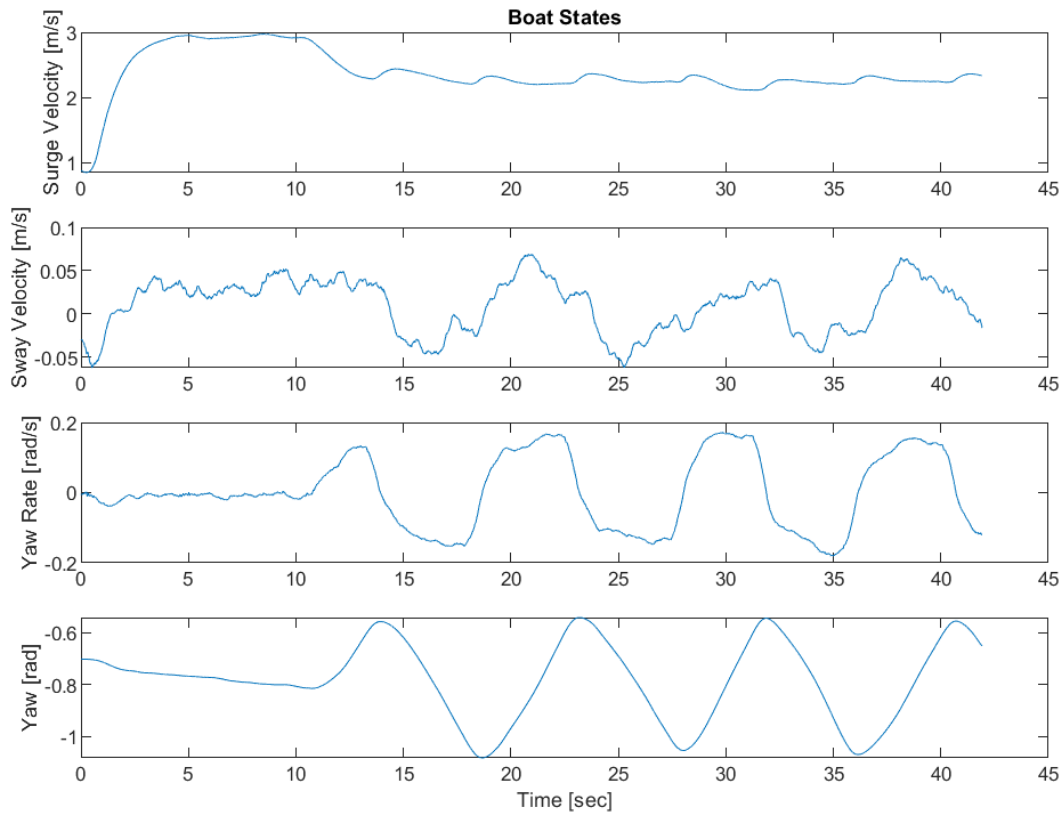


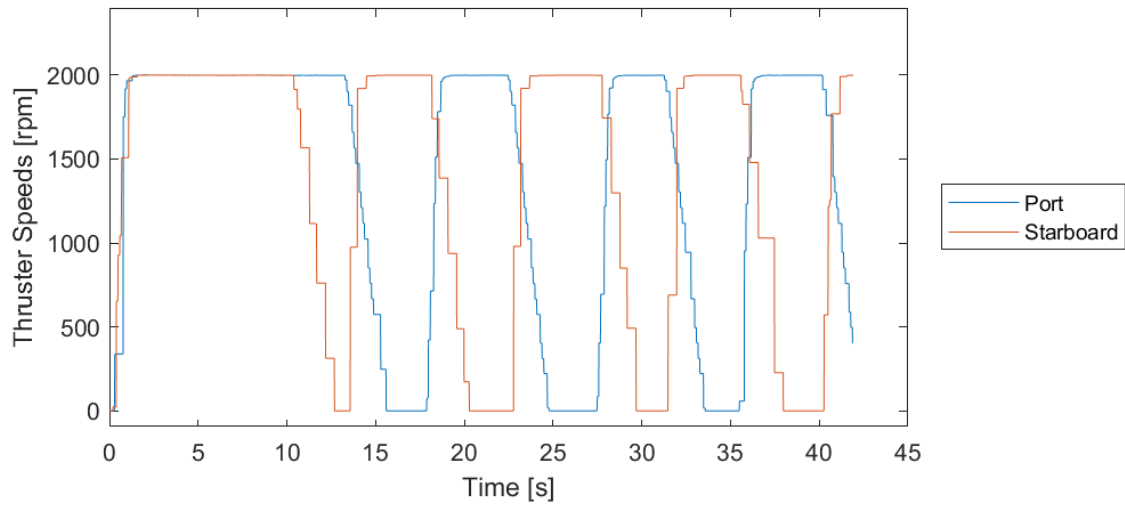
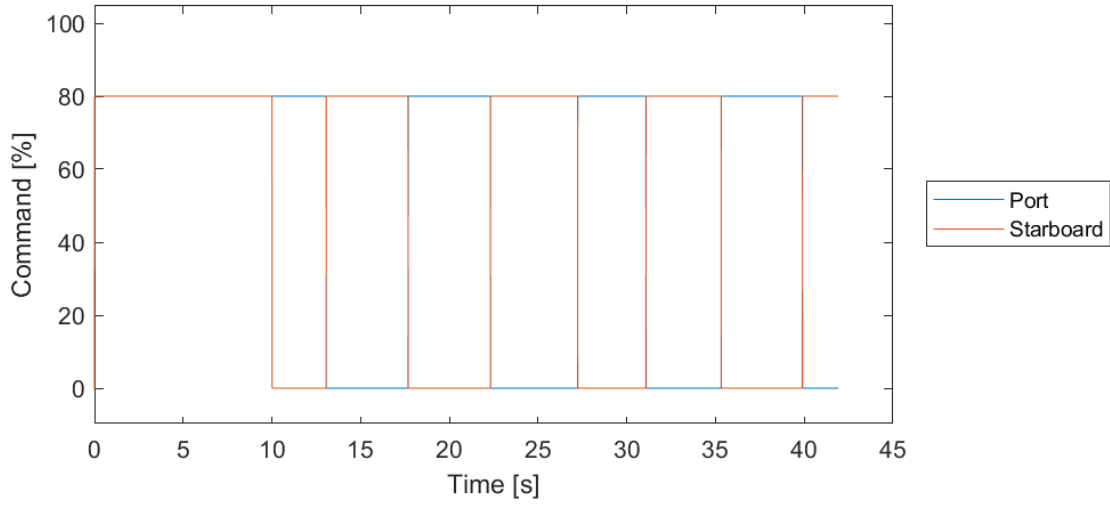
1.5.4. Zig-Zag Set 4



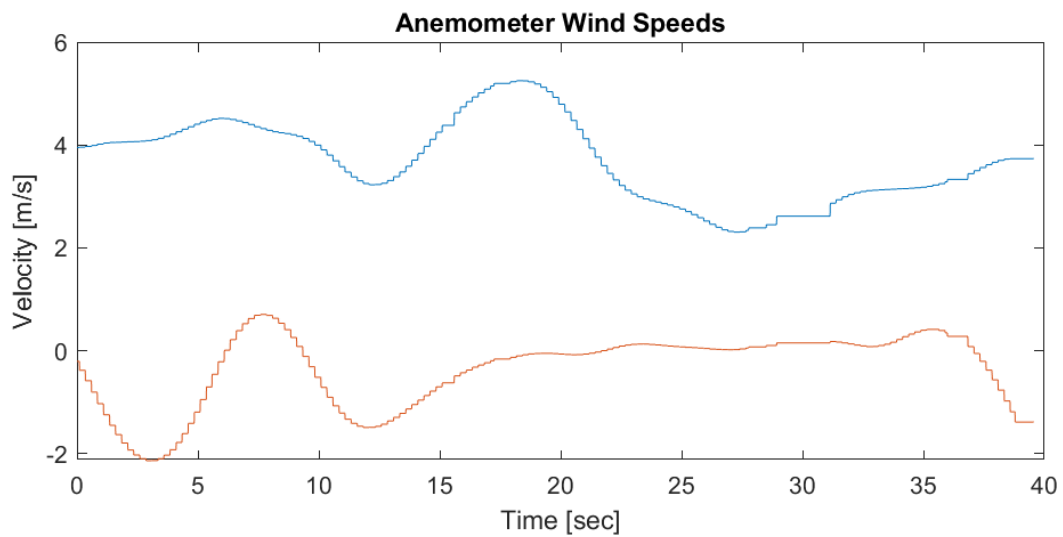
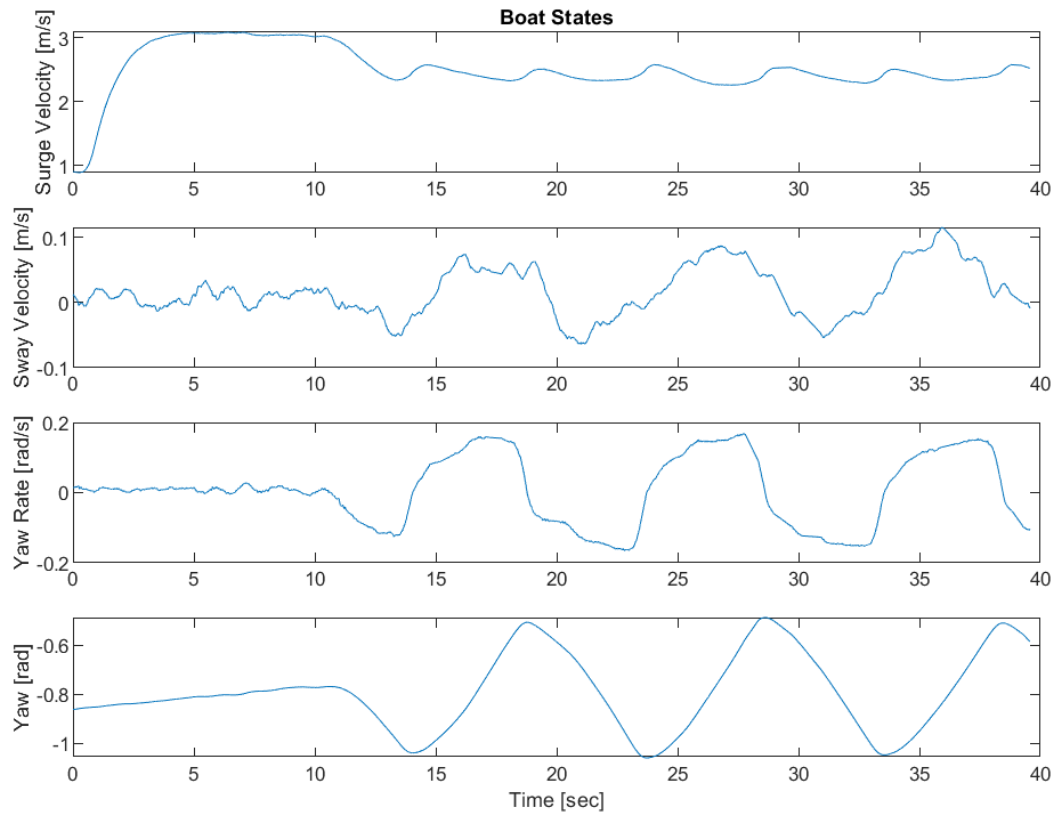


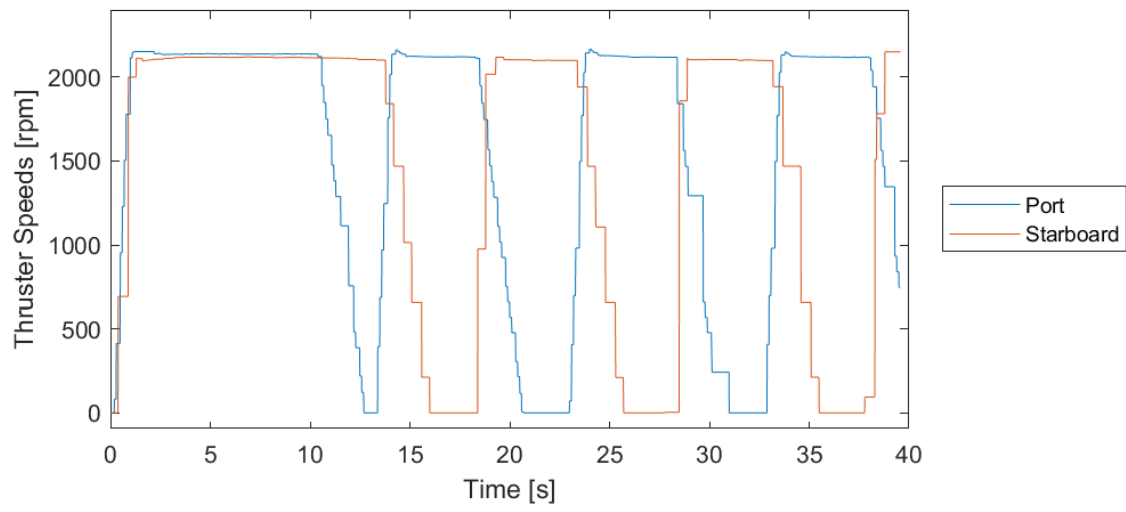
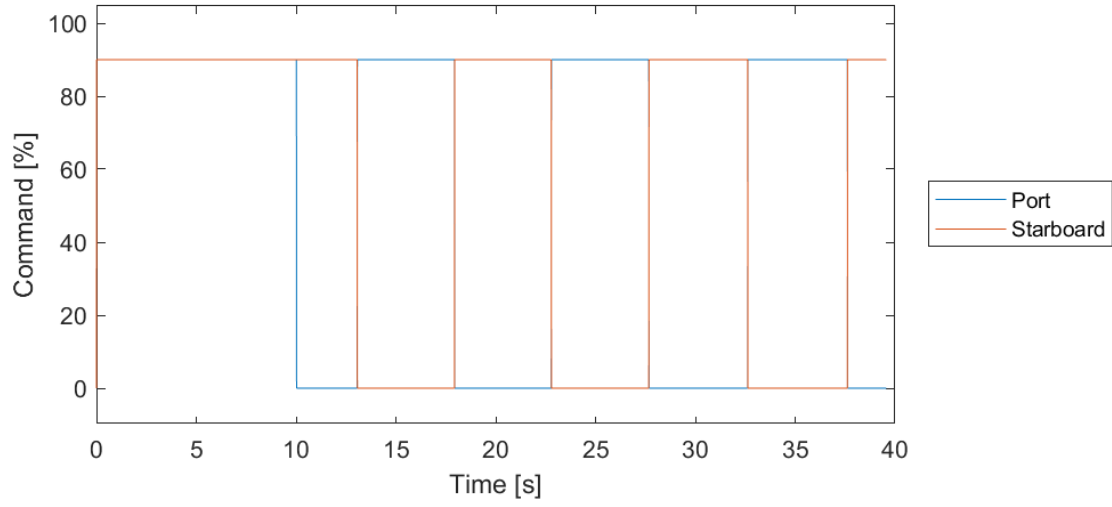
1.5.5. Zig-Zag Set 5



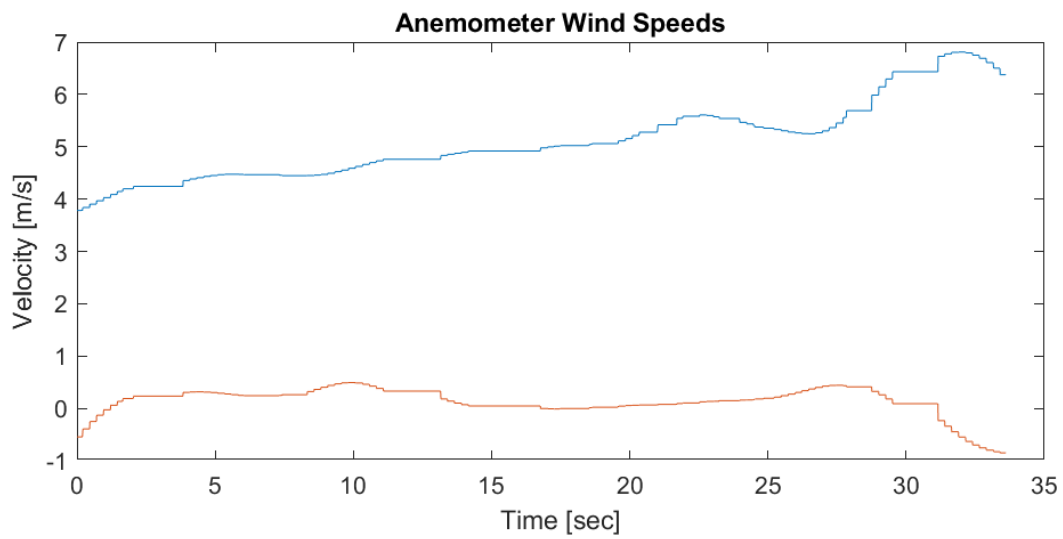
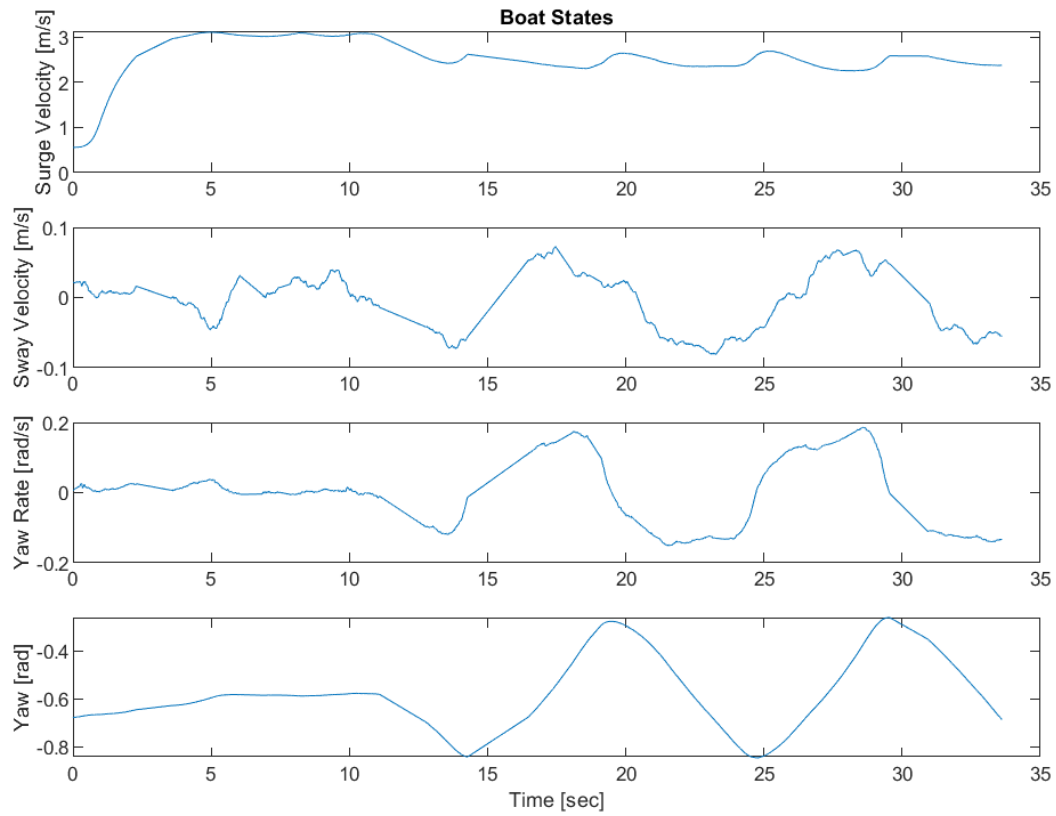


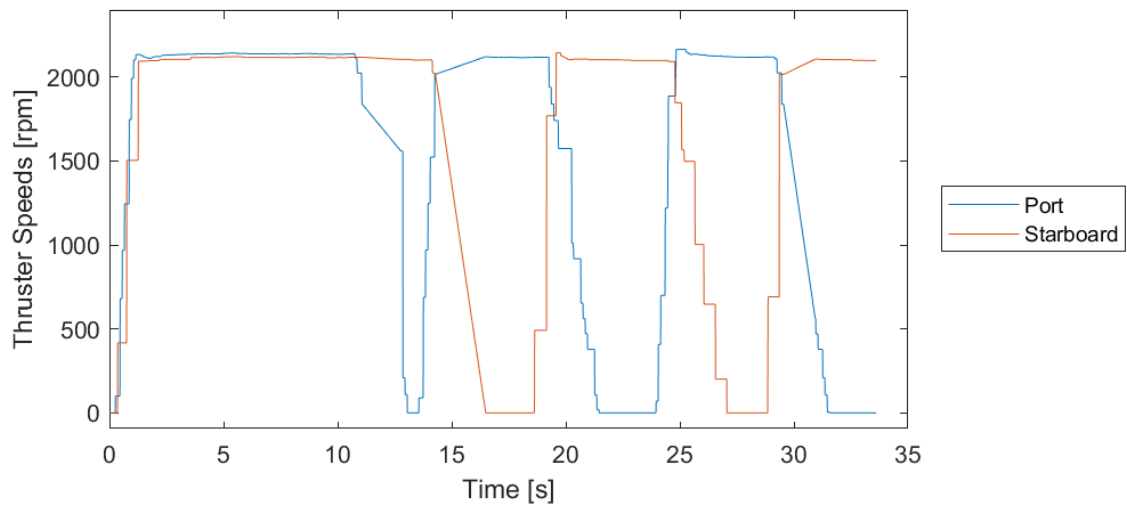
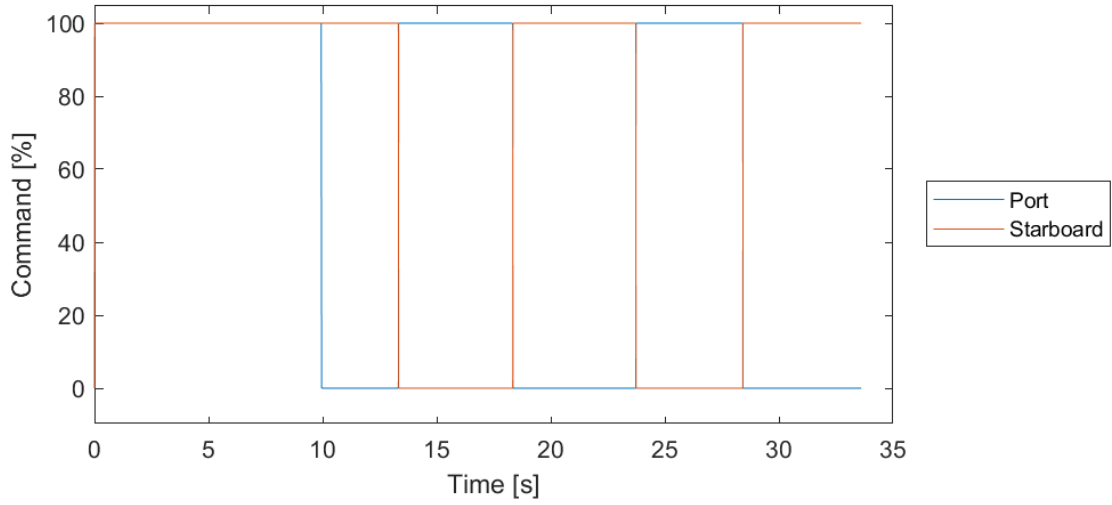
1.5.6. Zig-Zag Set 6



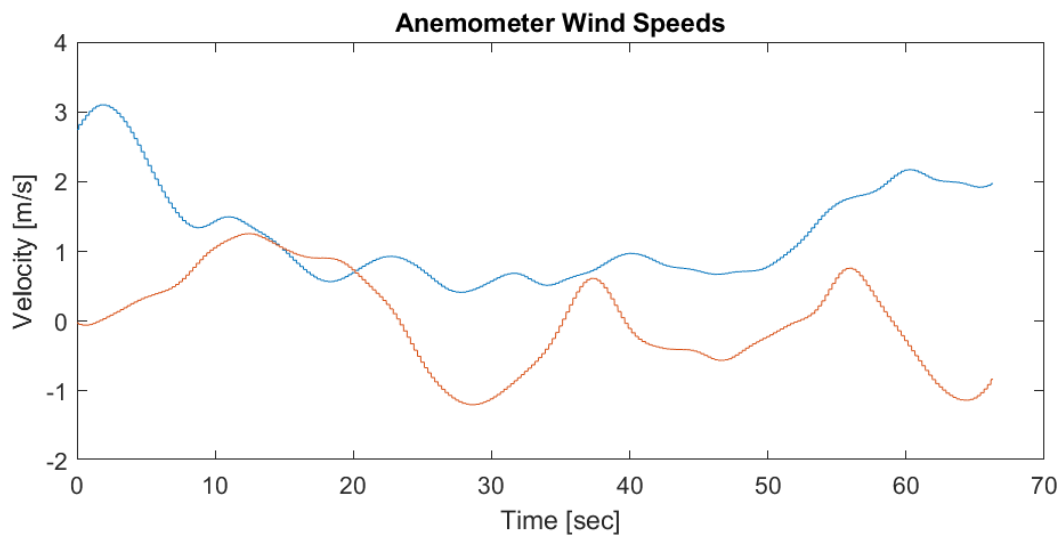
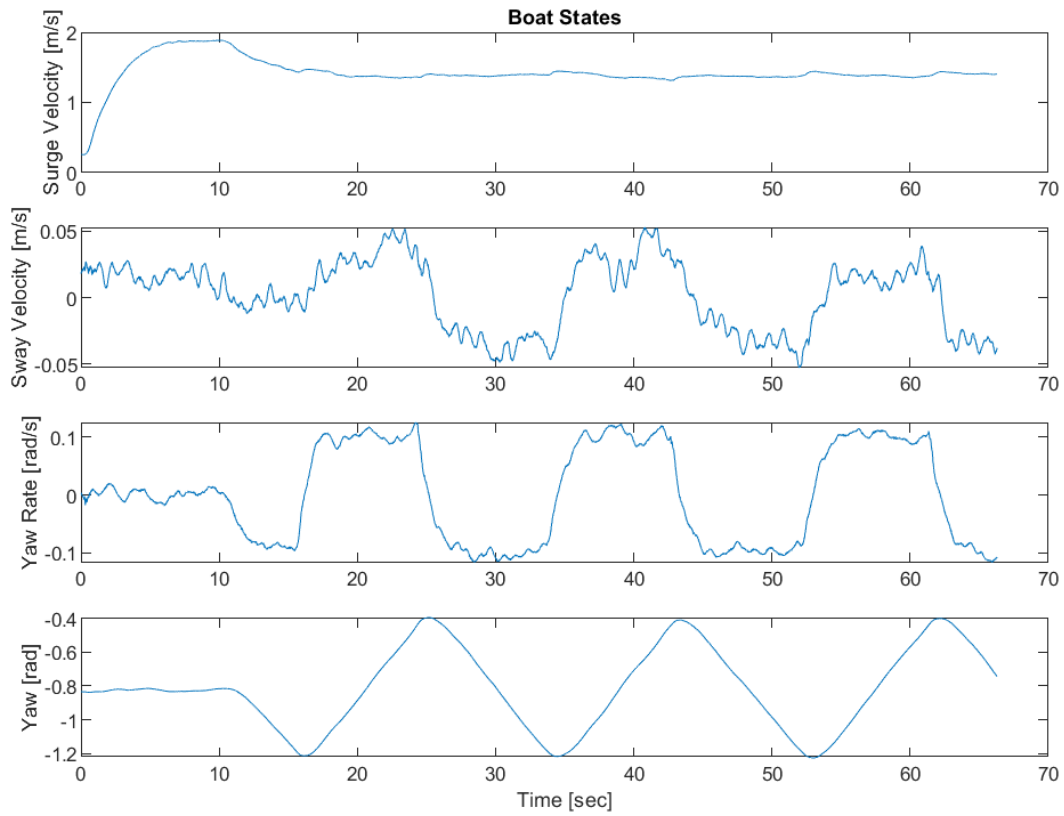


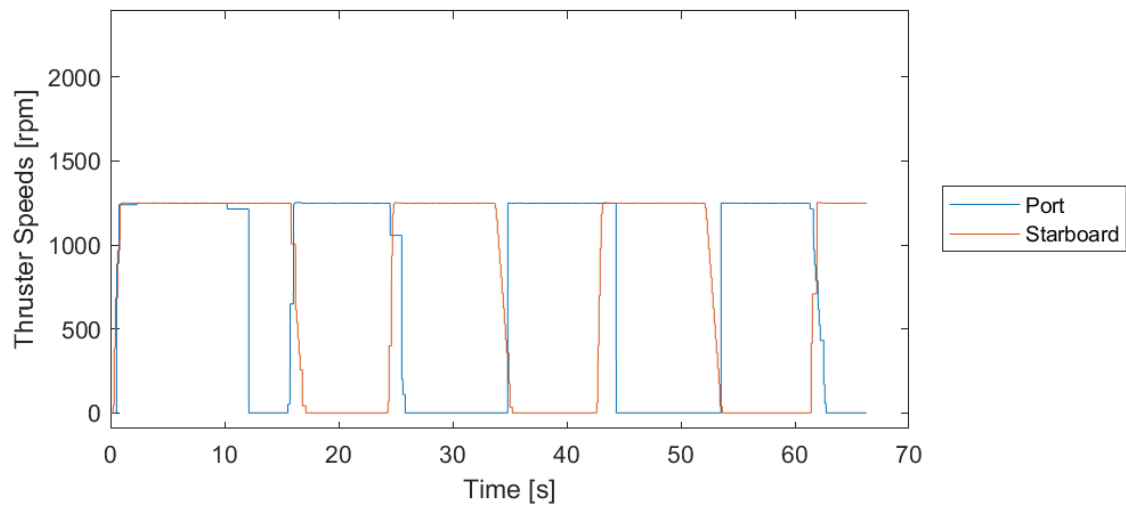
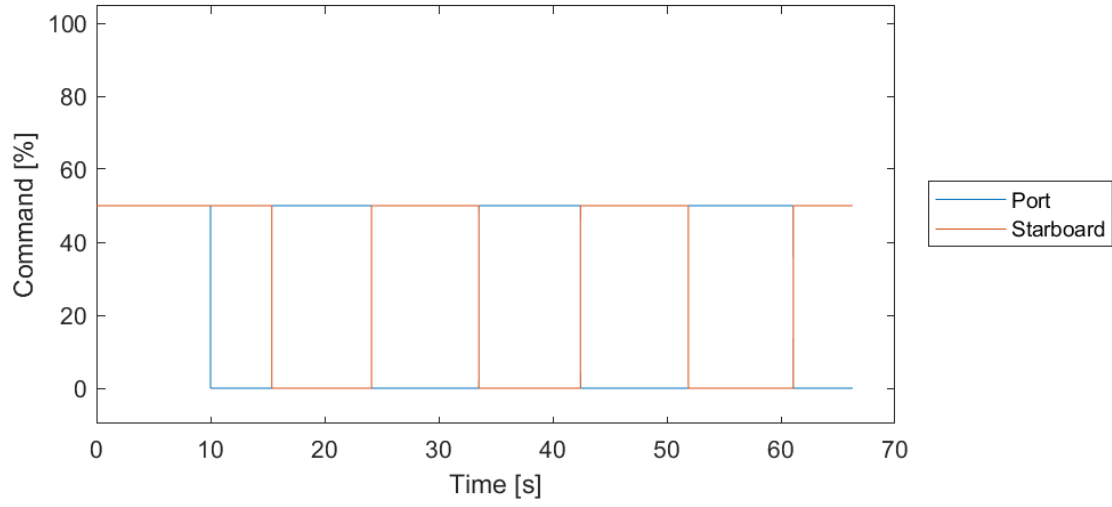
1.5.7. Zig-Zag Set 7



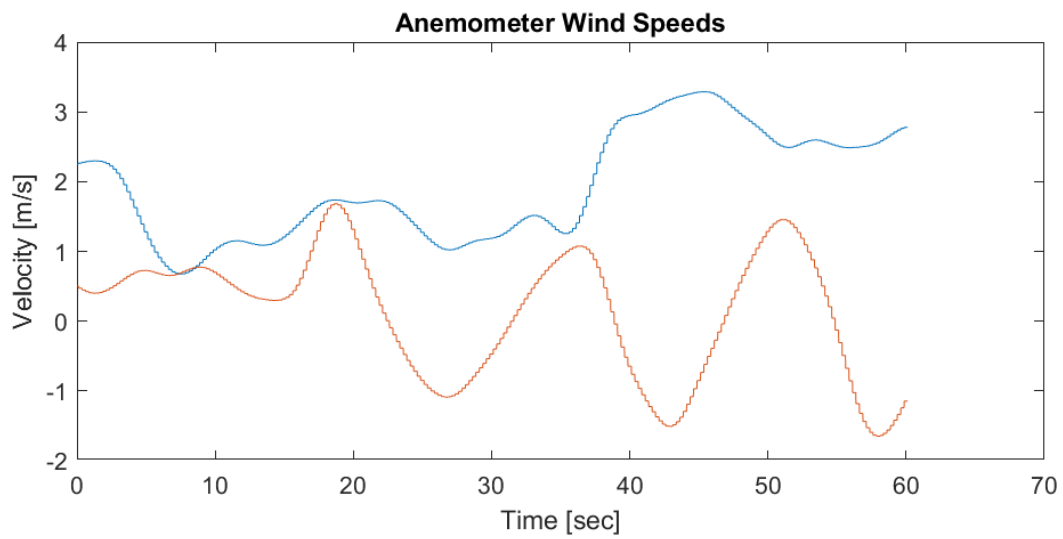
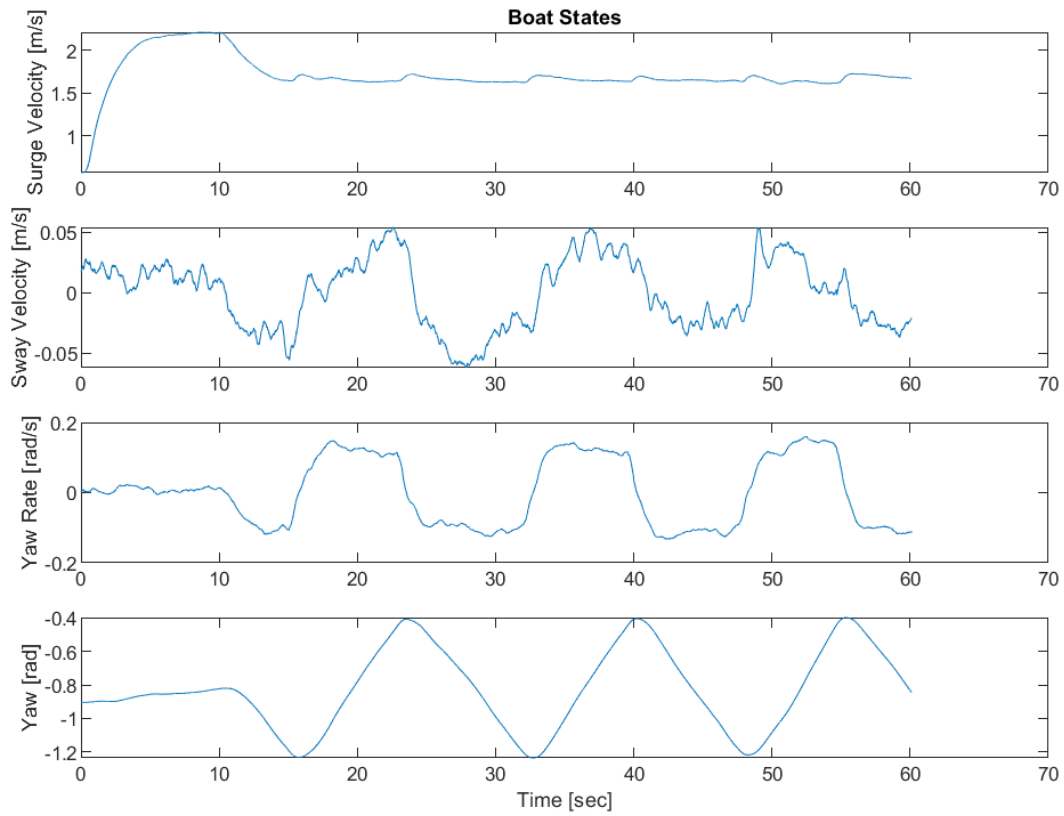


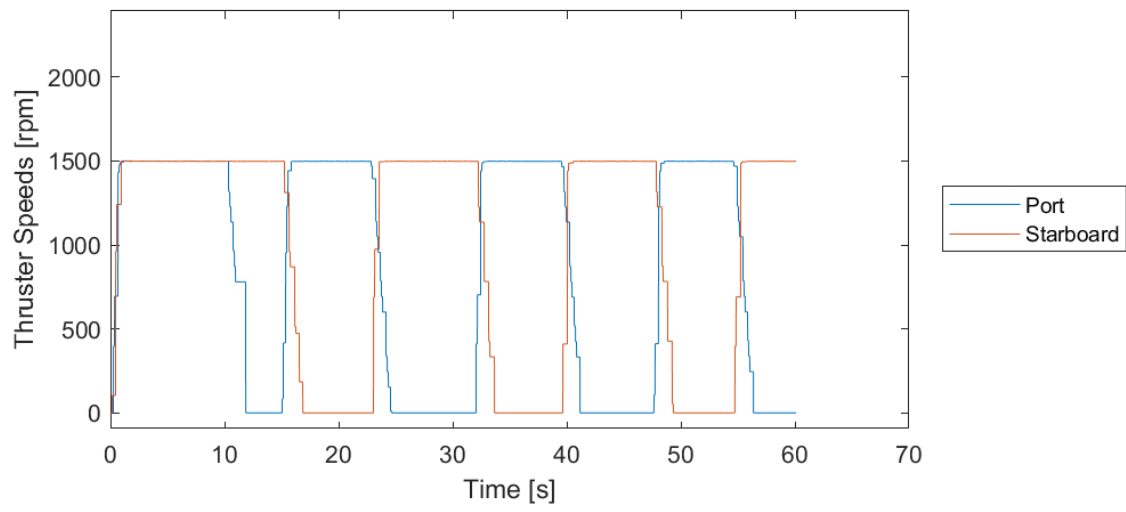
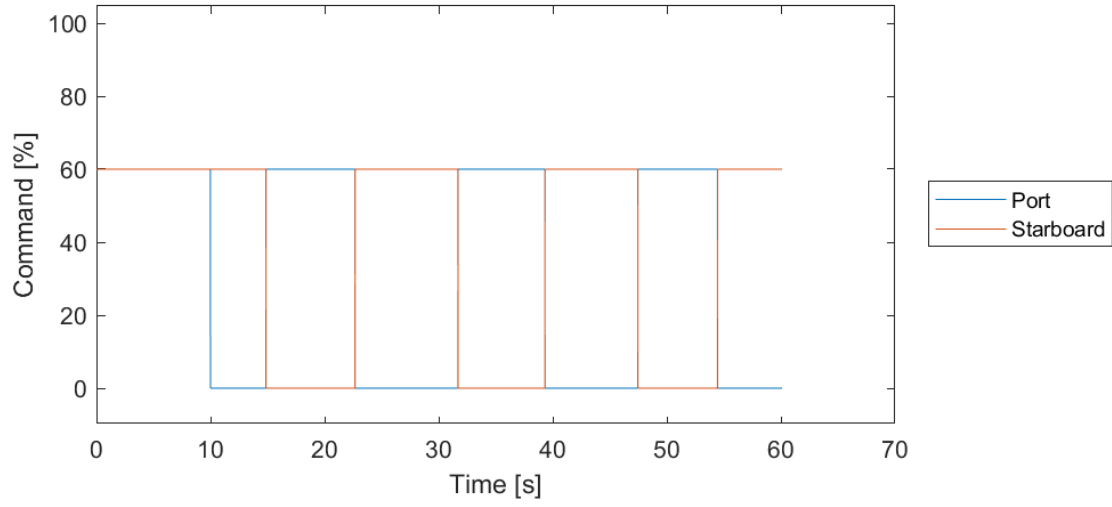
1.5.8. Zig-Zag Set 8



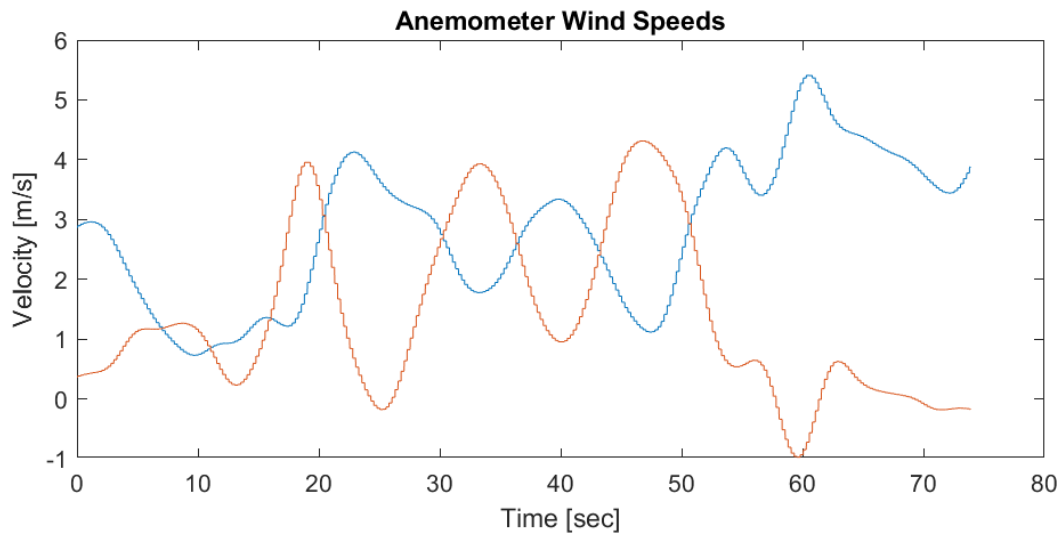
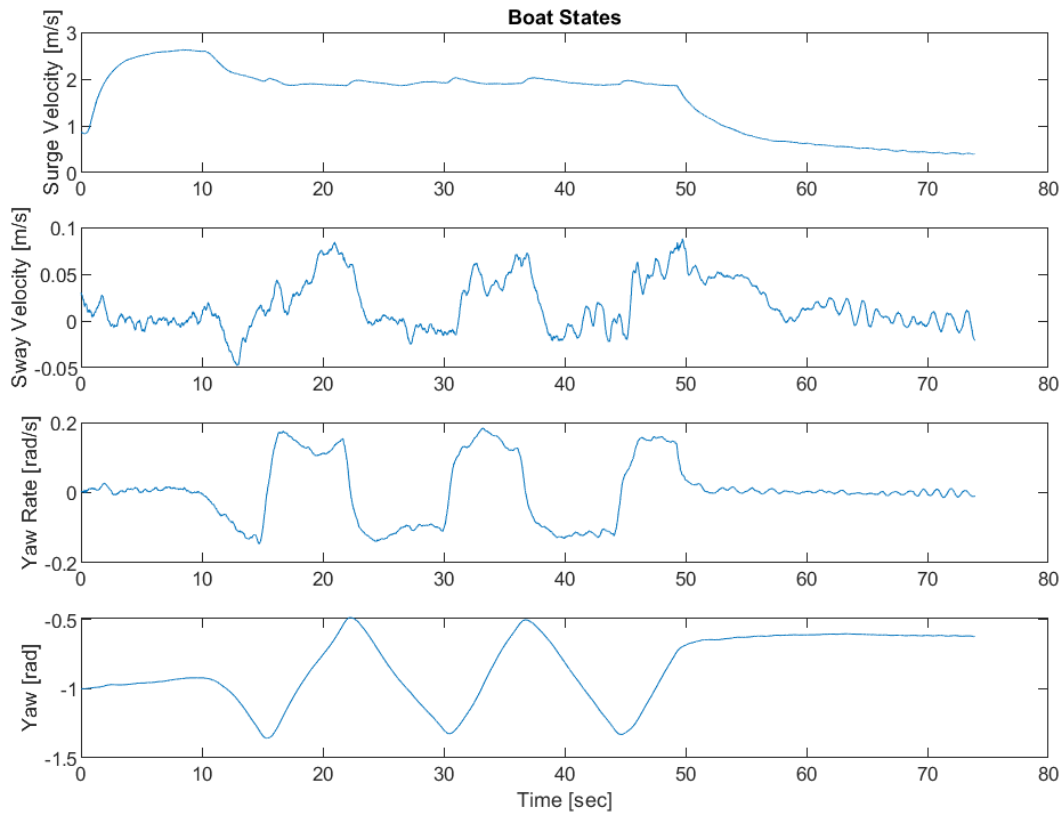


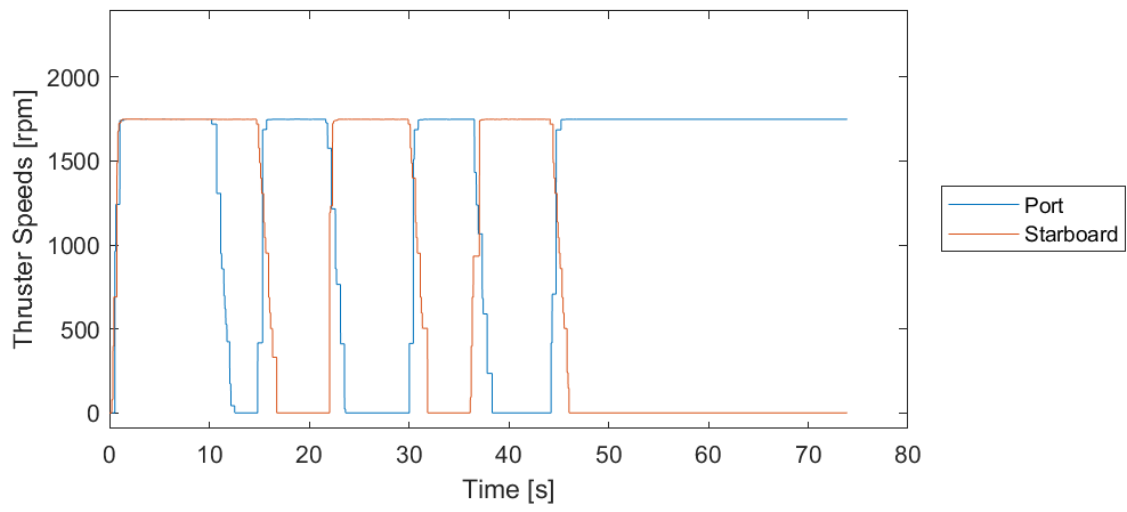
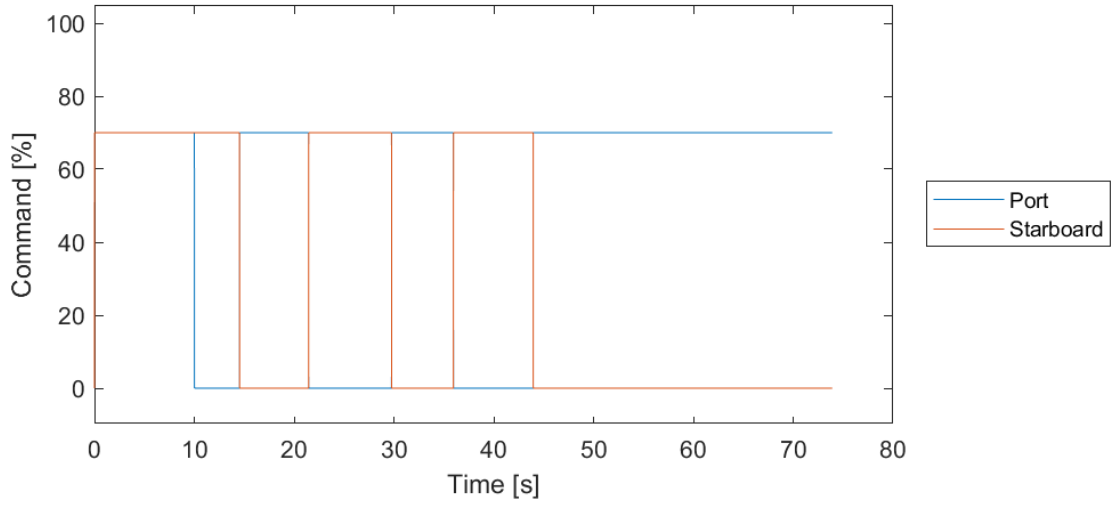
1.5.9. Zig-Zag Set 9



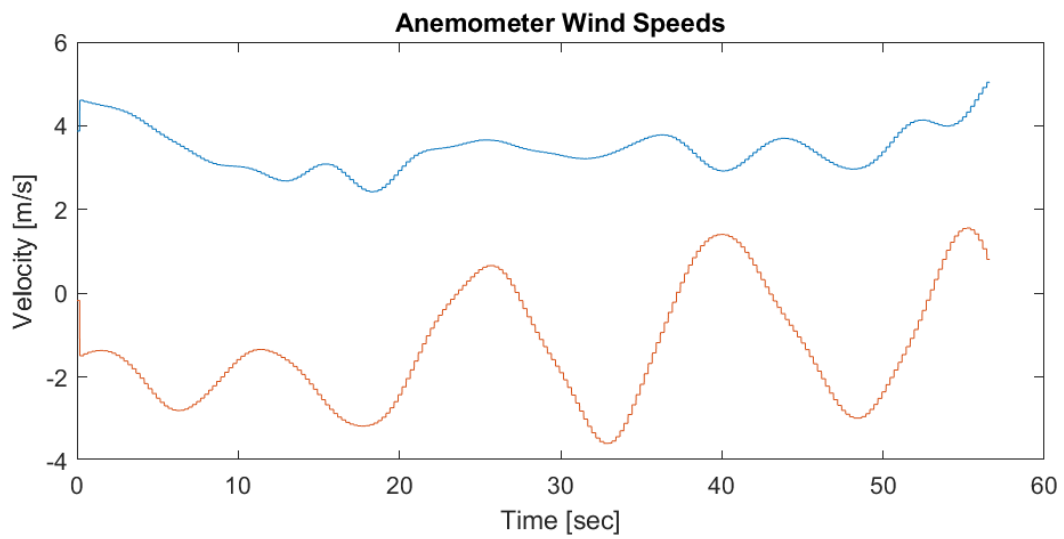
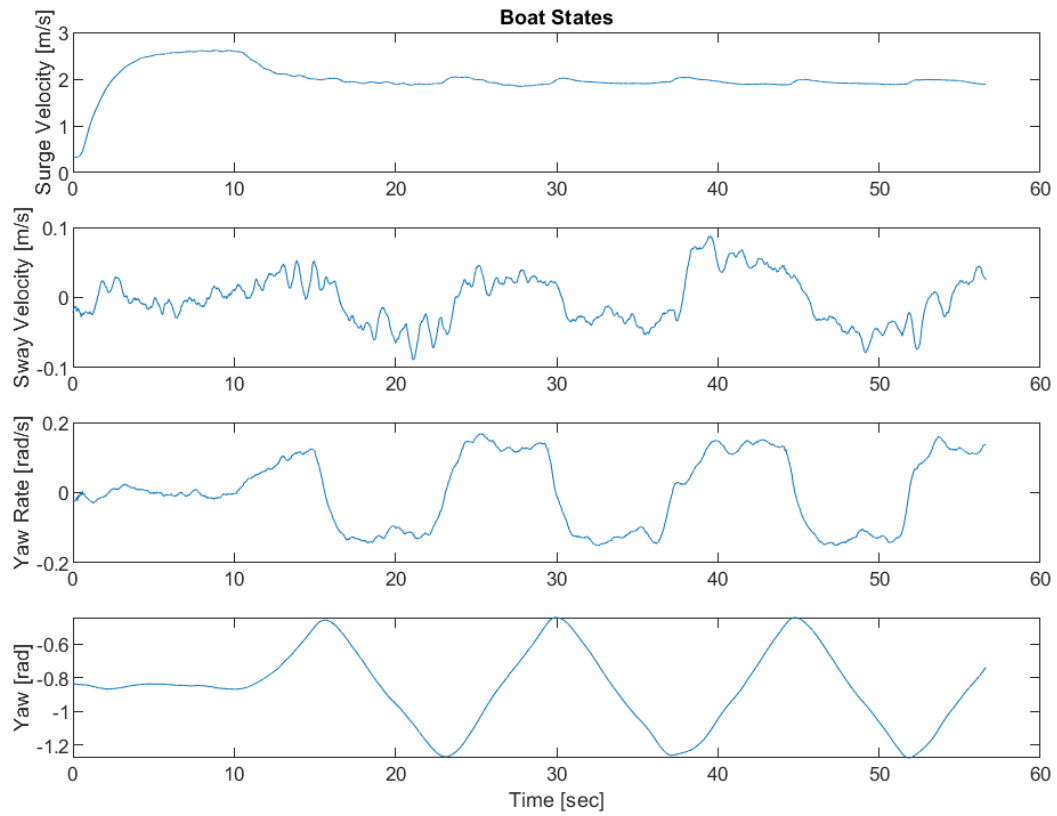


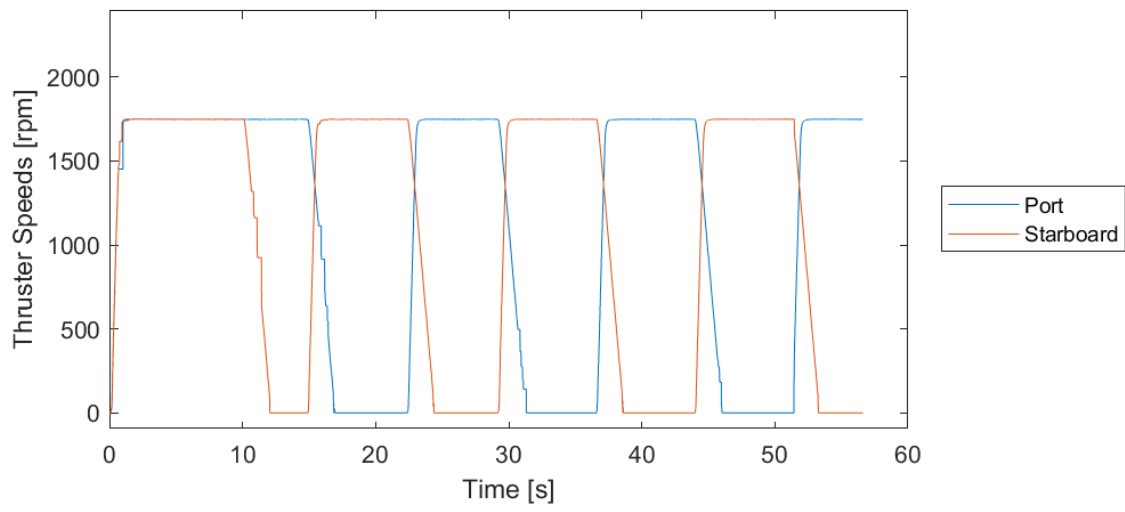
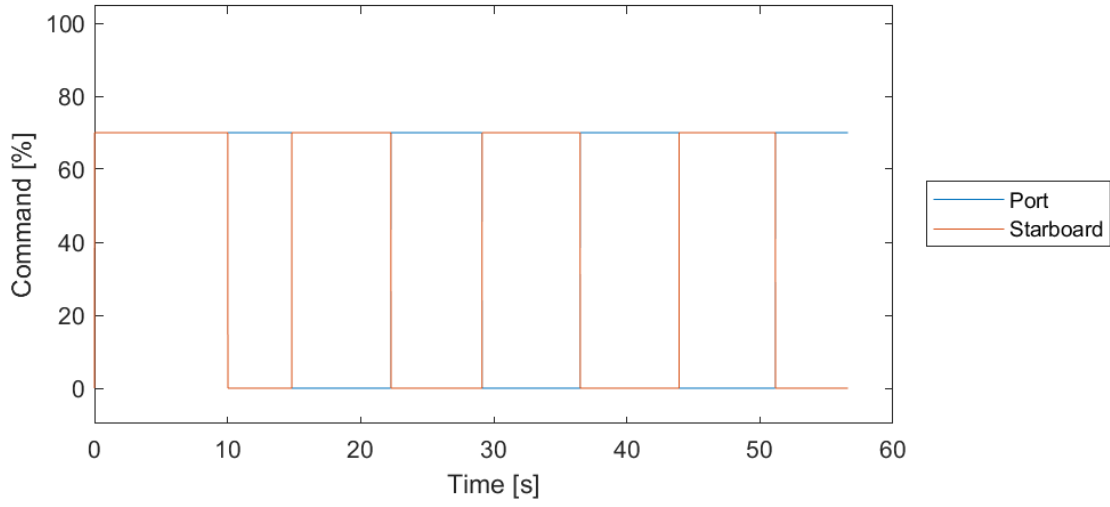
1.5.10. Zig-Zag Set 10



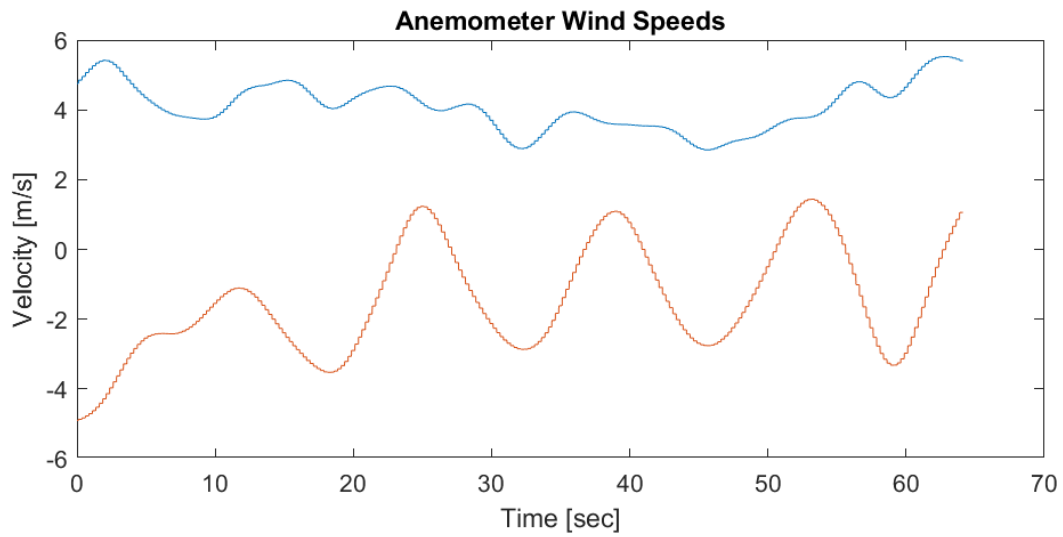
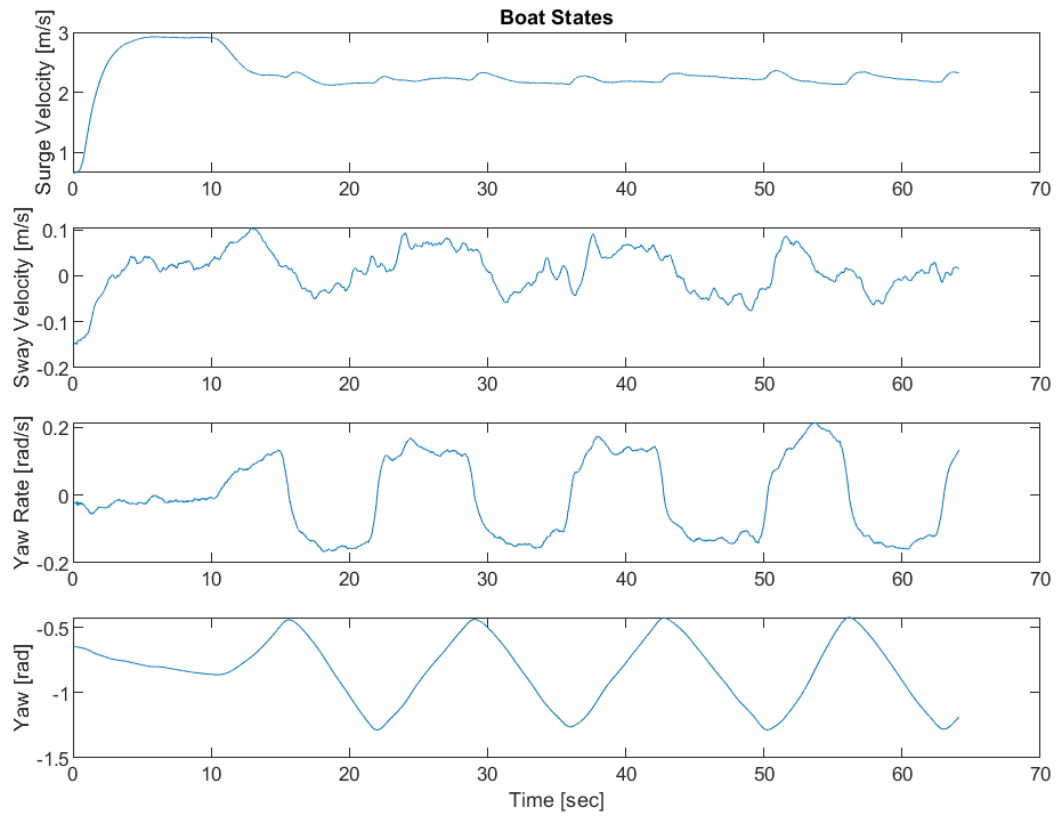


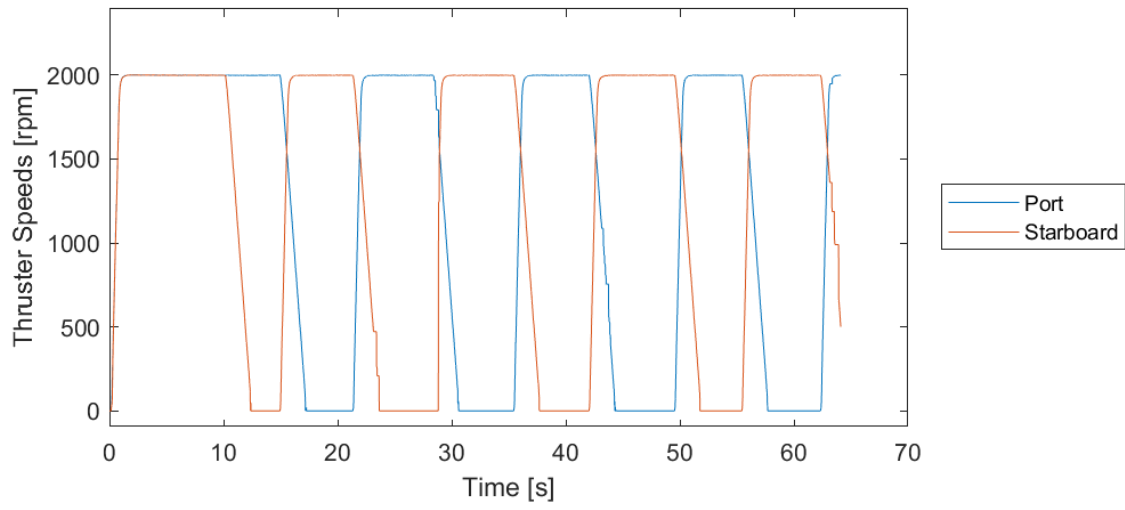
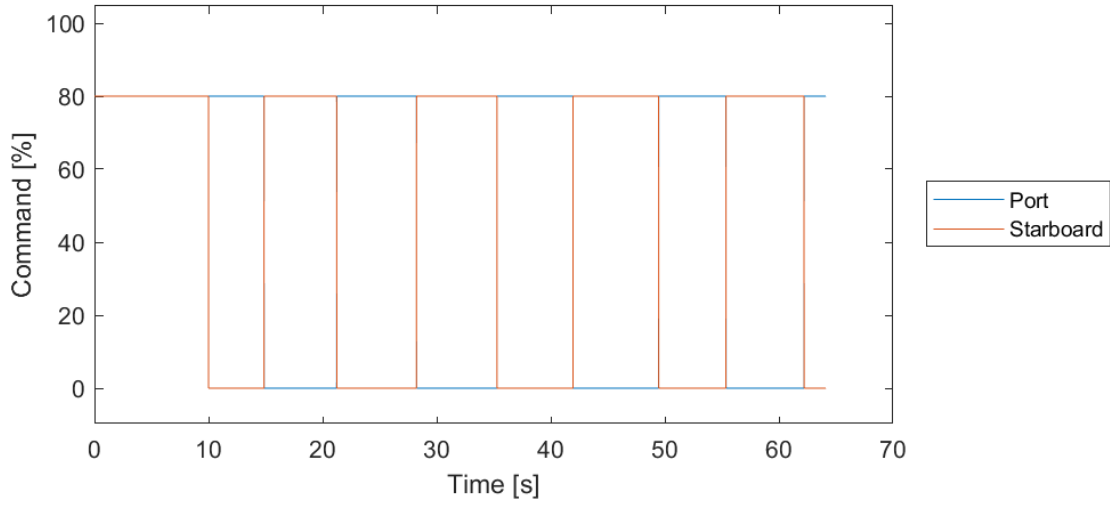
1.5.11. Zig-Zag Set 11



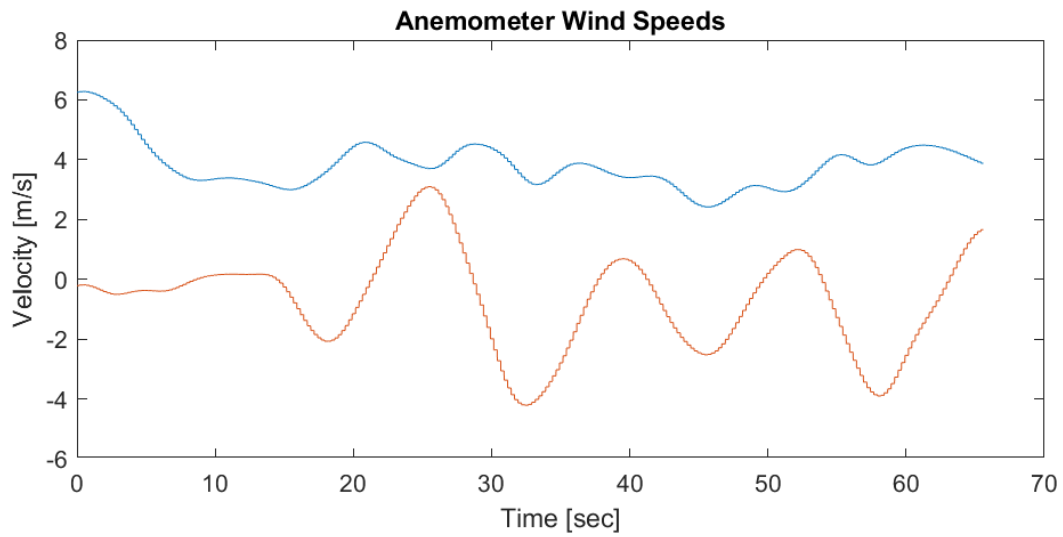
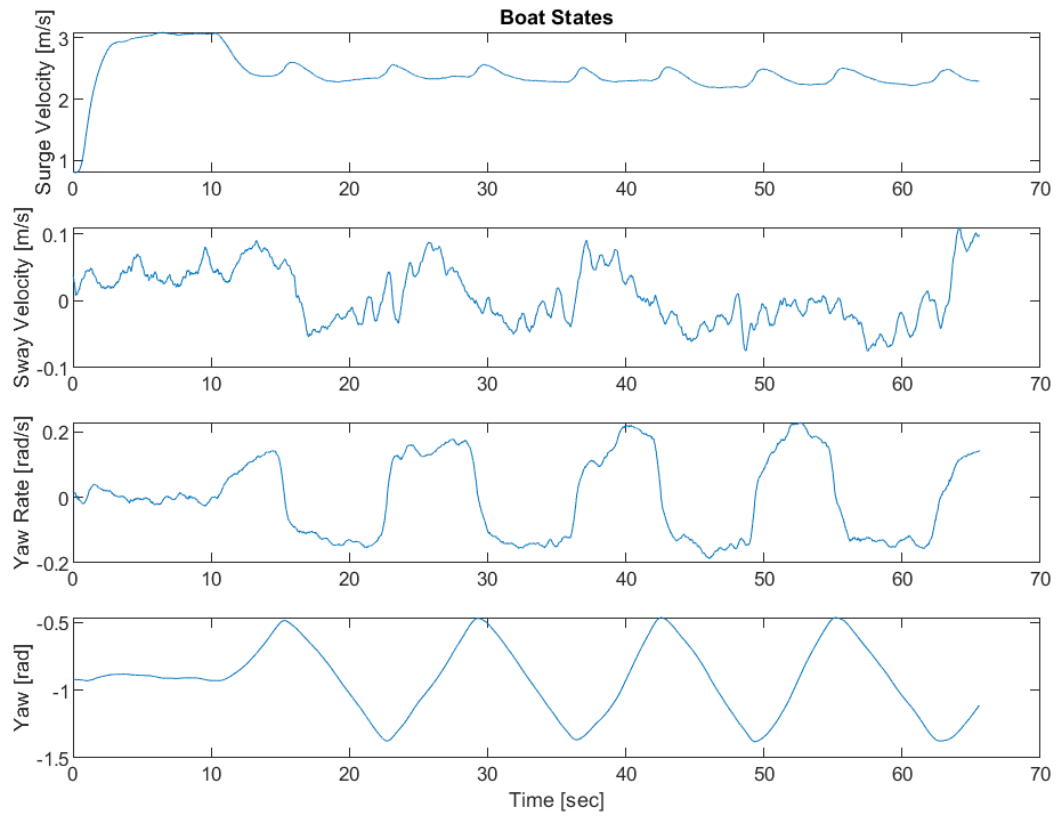


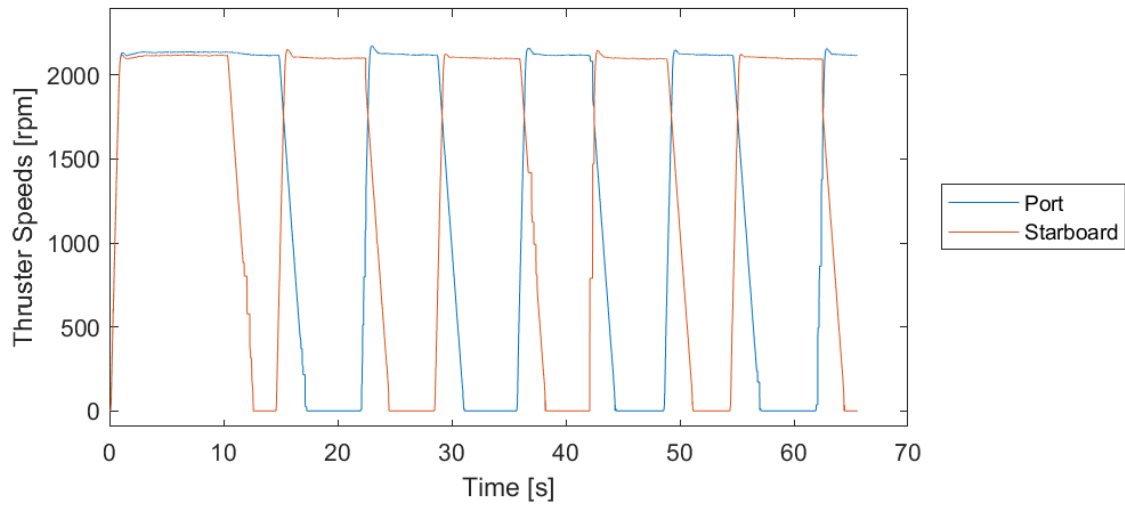
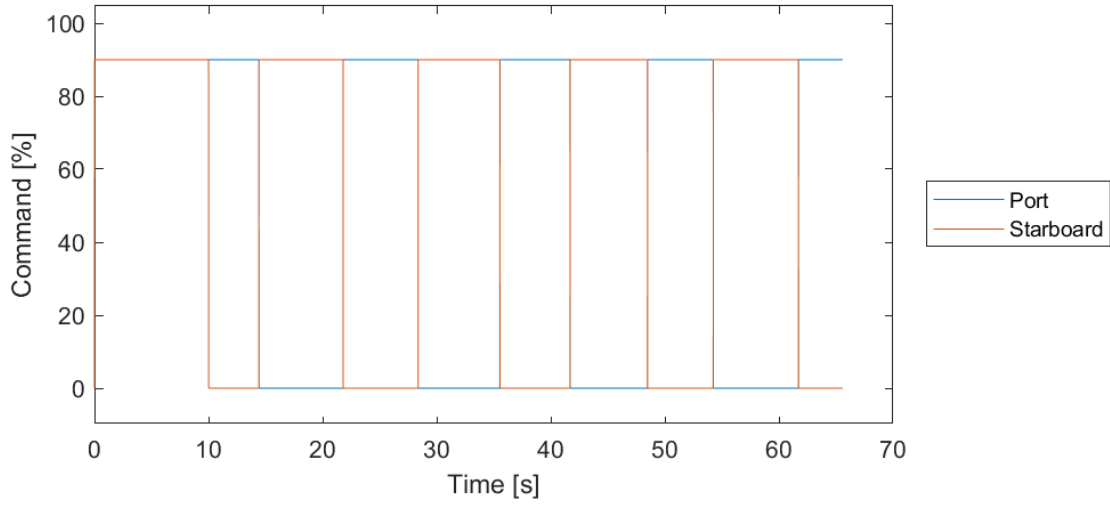
1.5.12. Zig-Zag Set 12



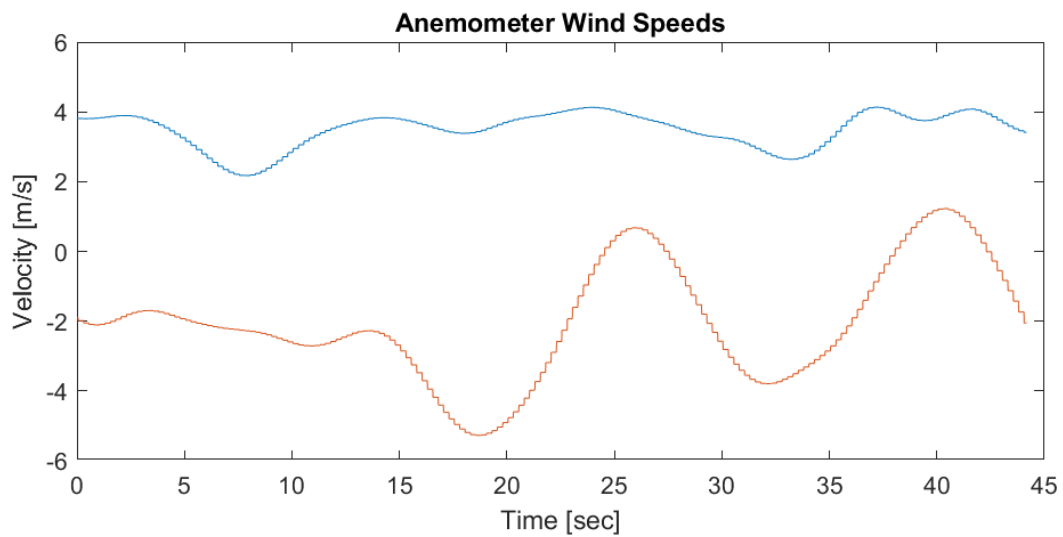
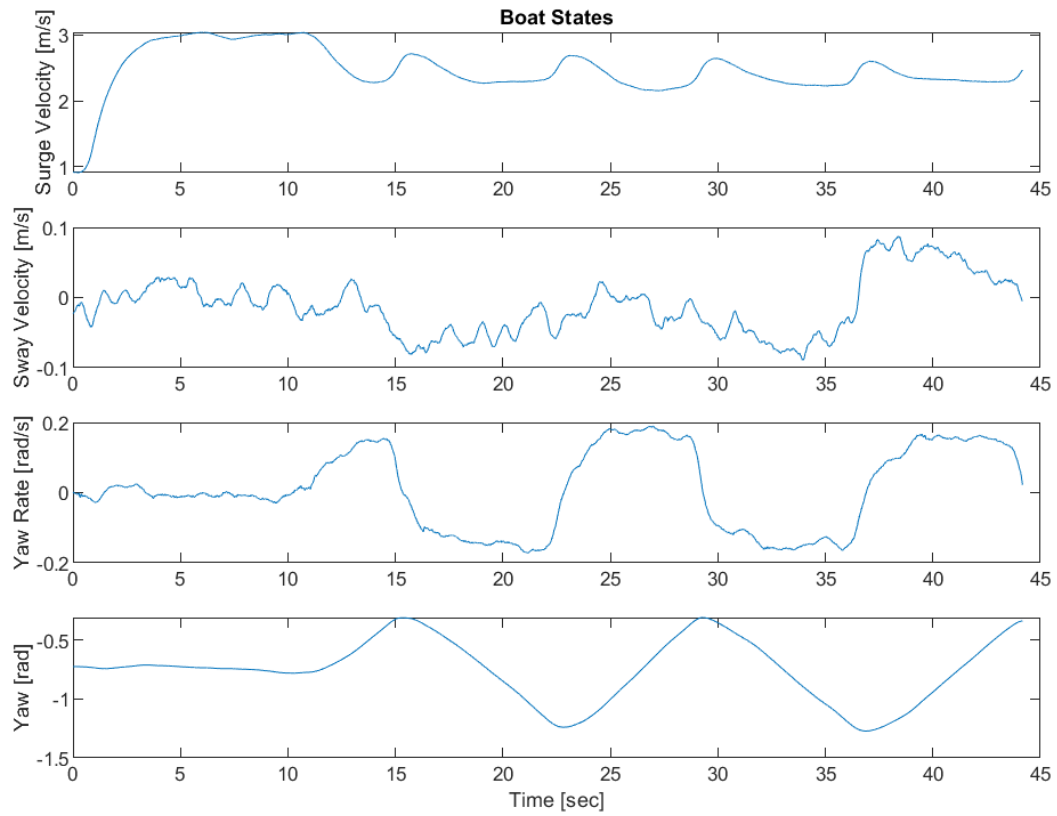


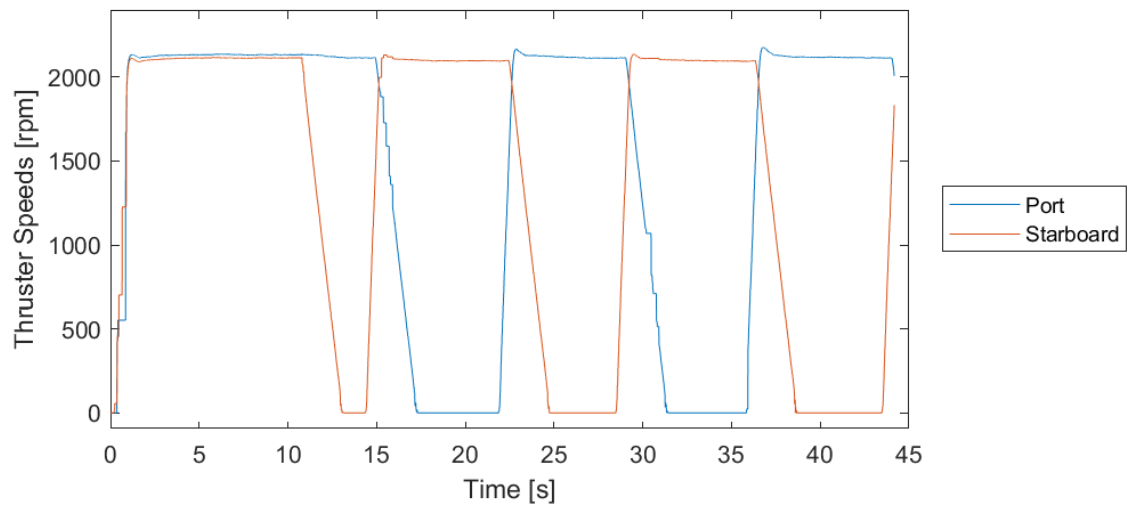
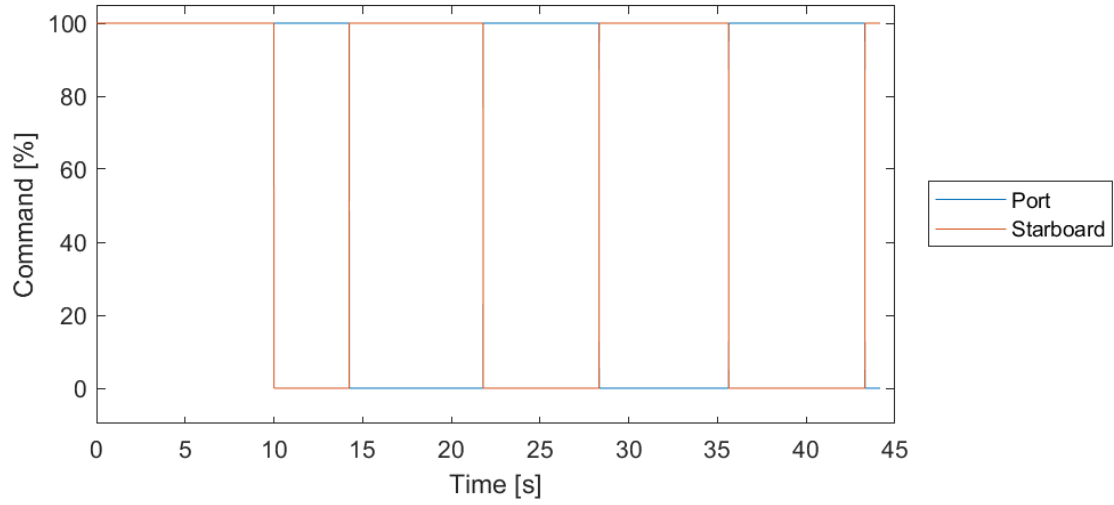
1.5.13. Zig-Zag Set 13

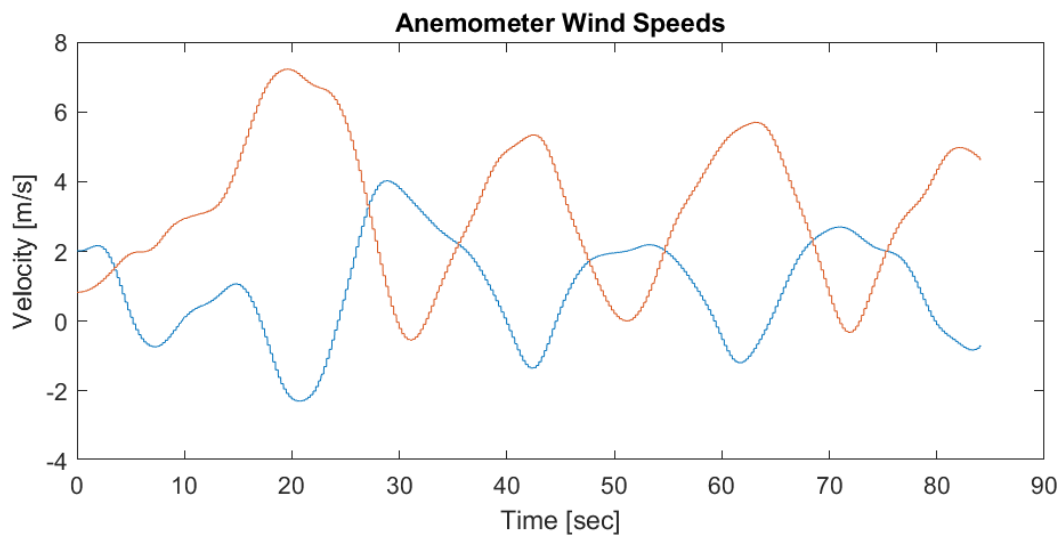
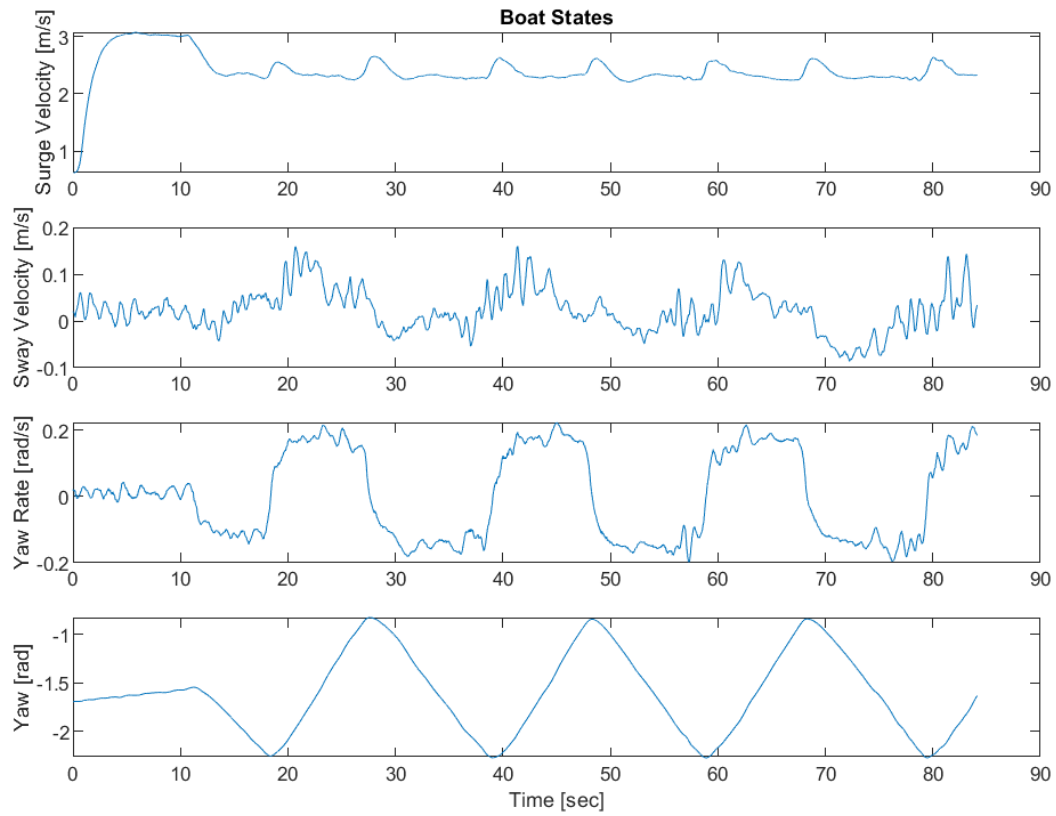


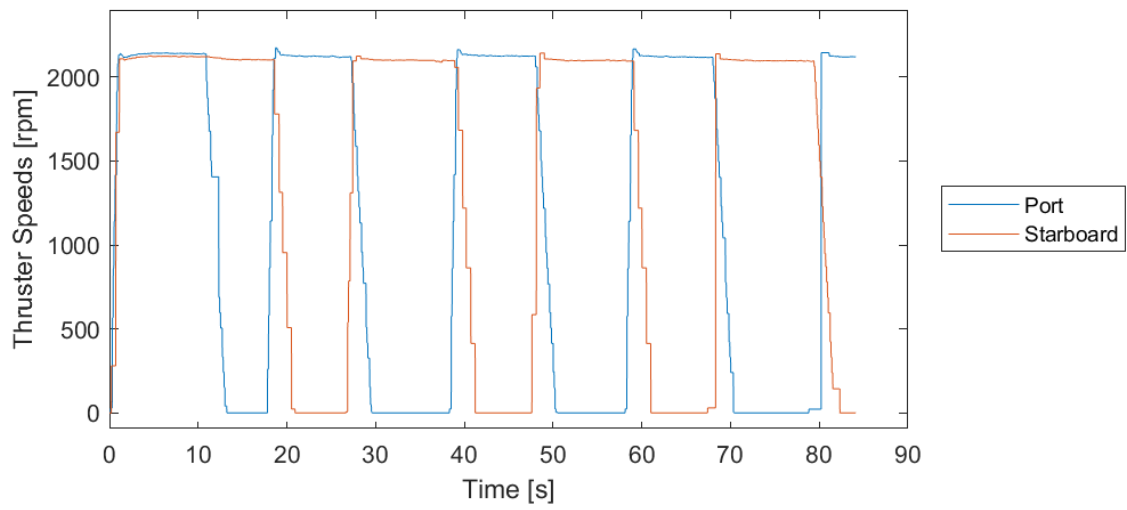
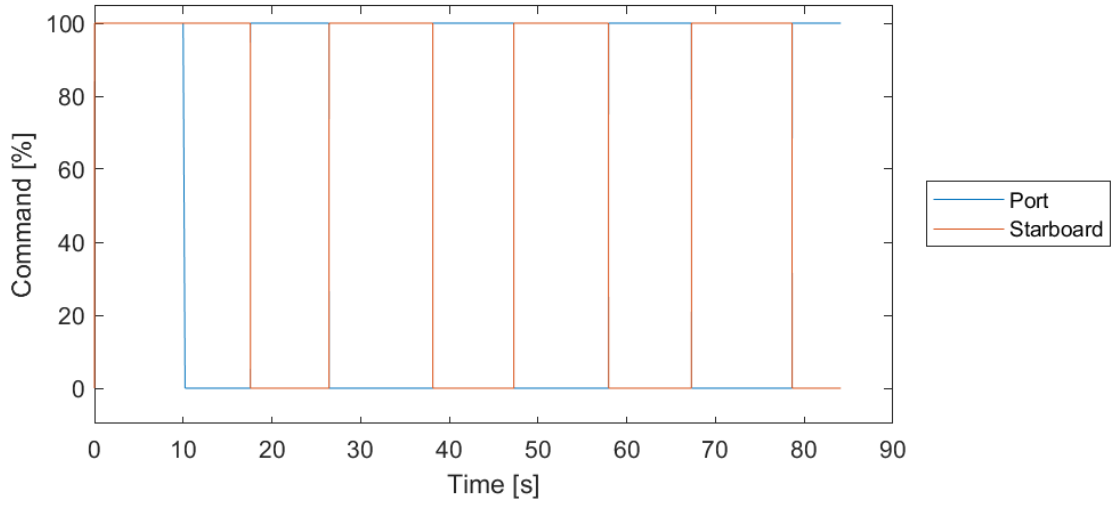


1.5.14. Zig-Zag Set 14

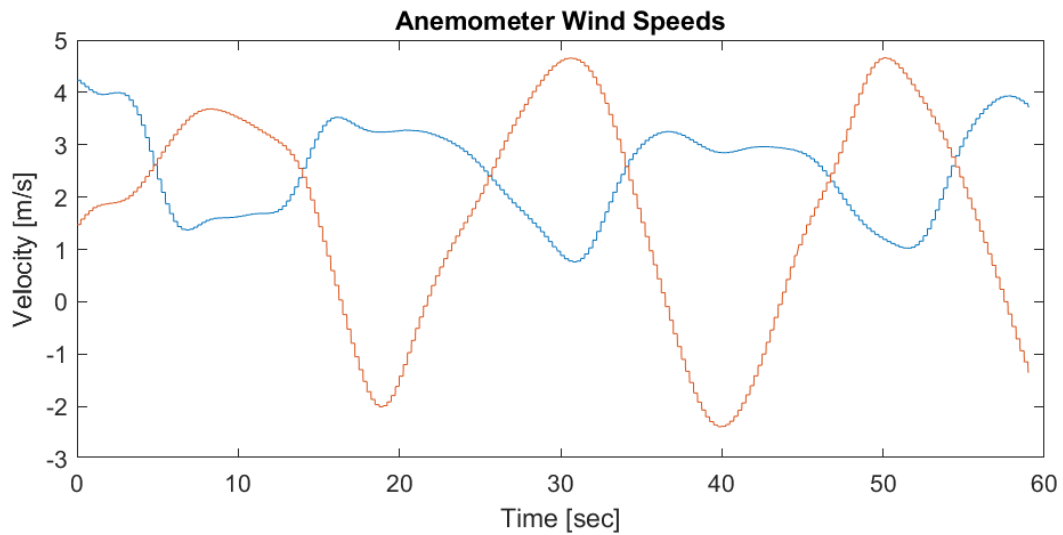
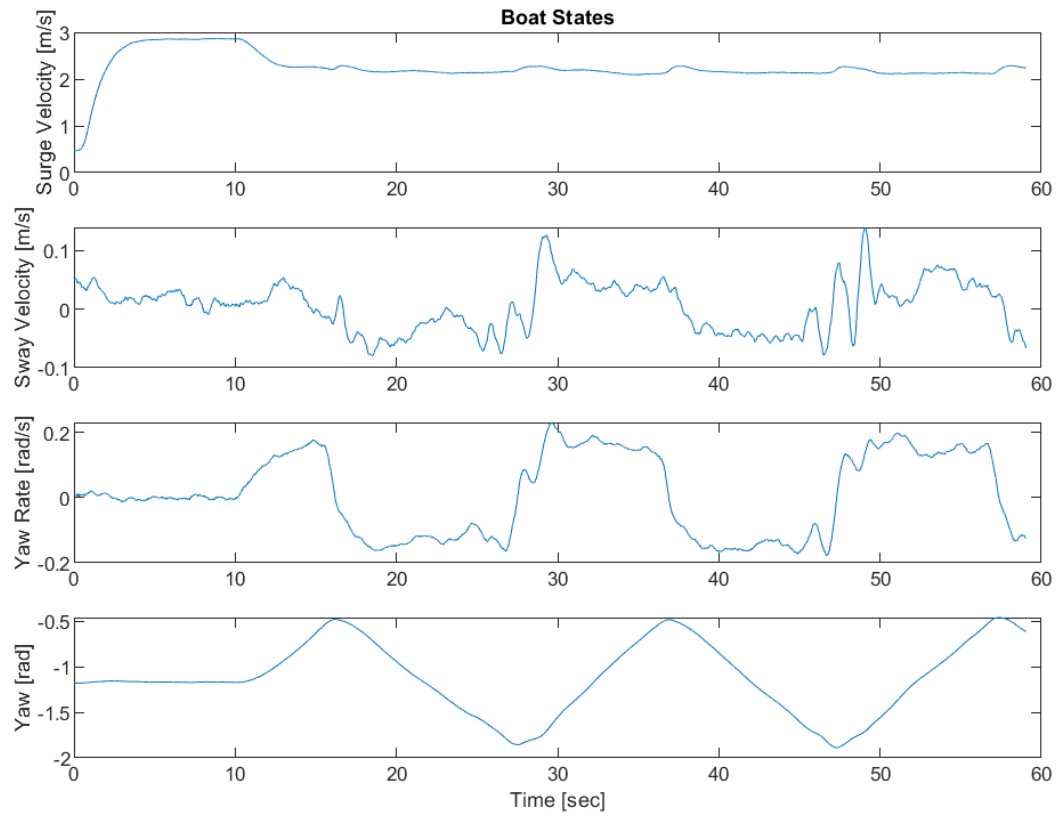


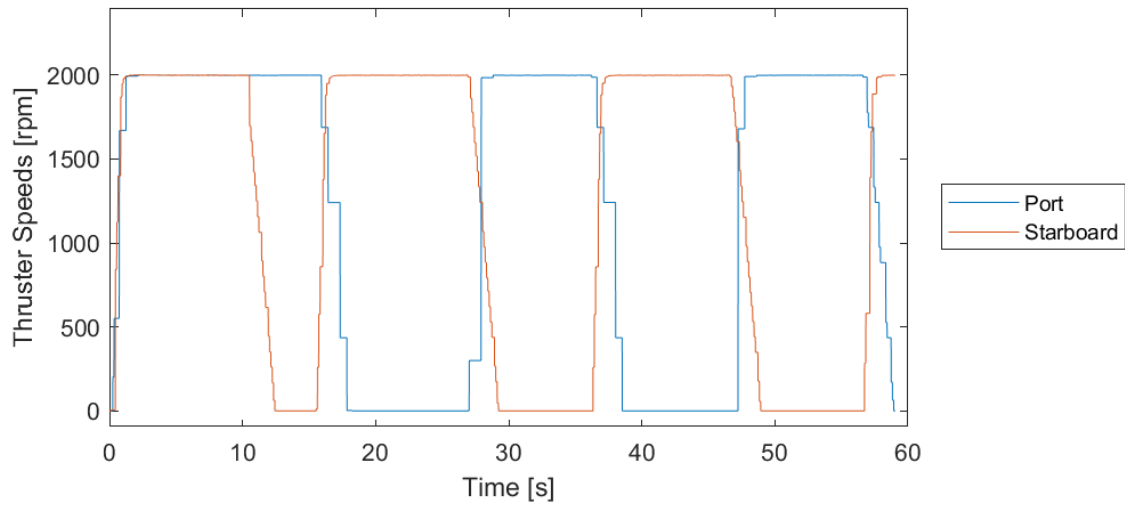
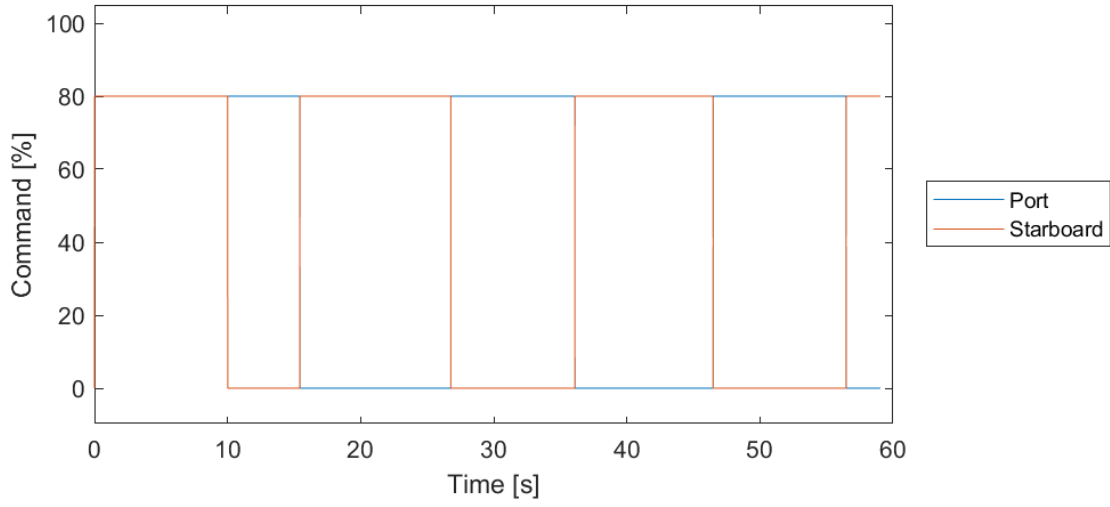


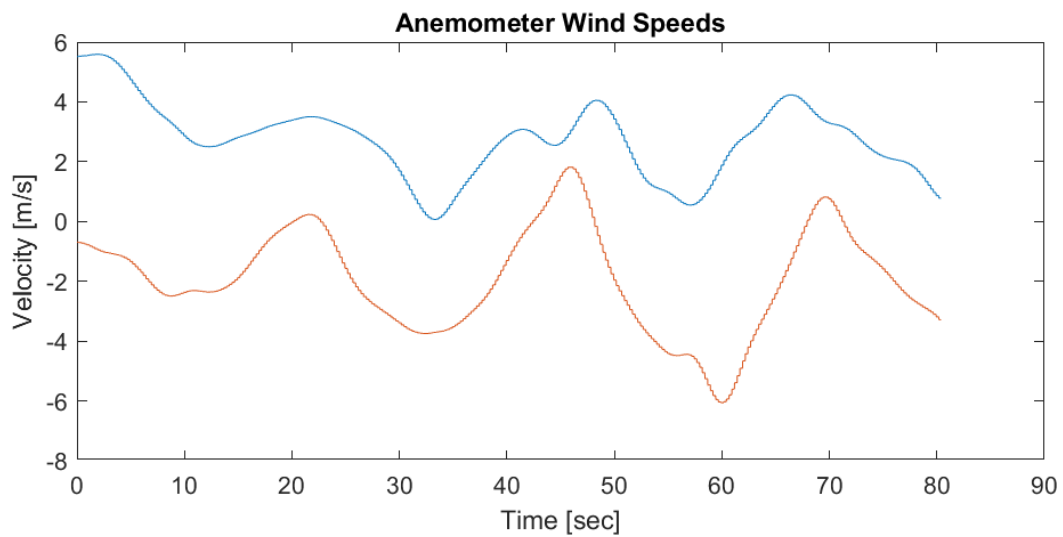
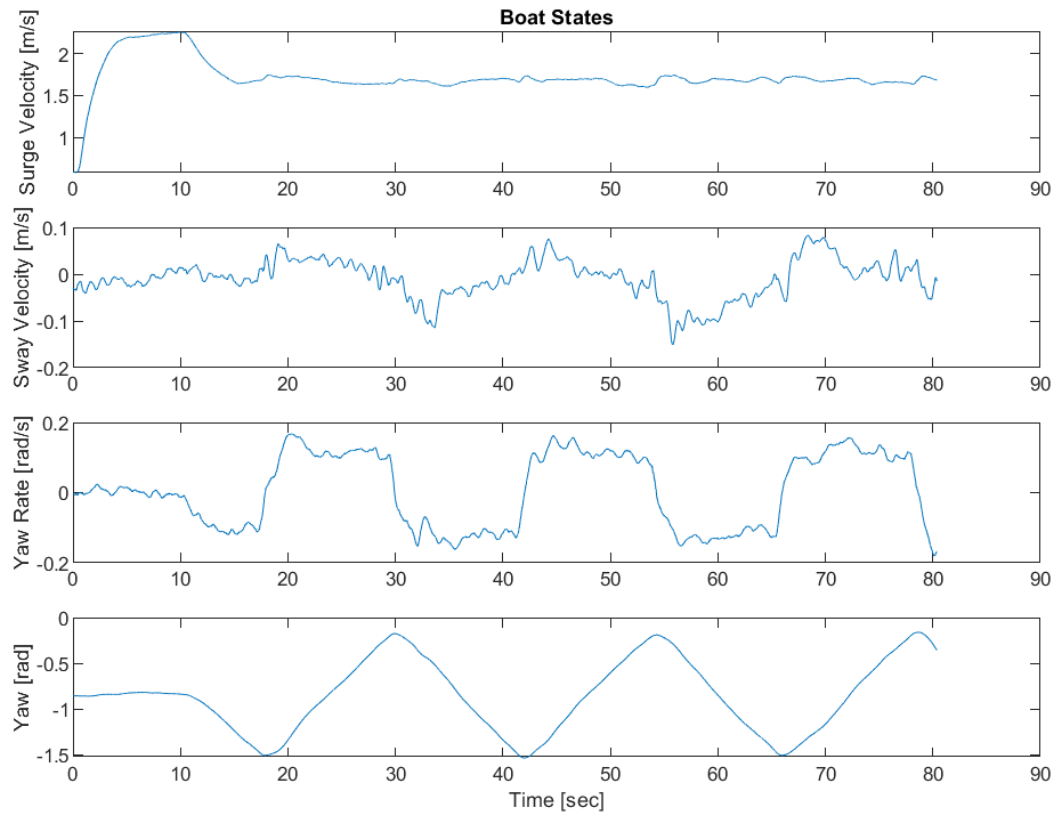
1.5.15. **Zig-Zag Set 15**

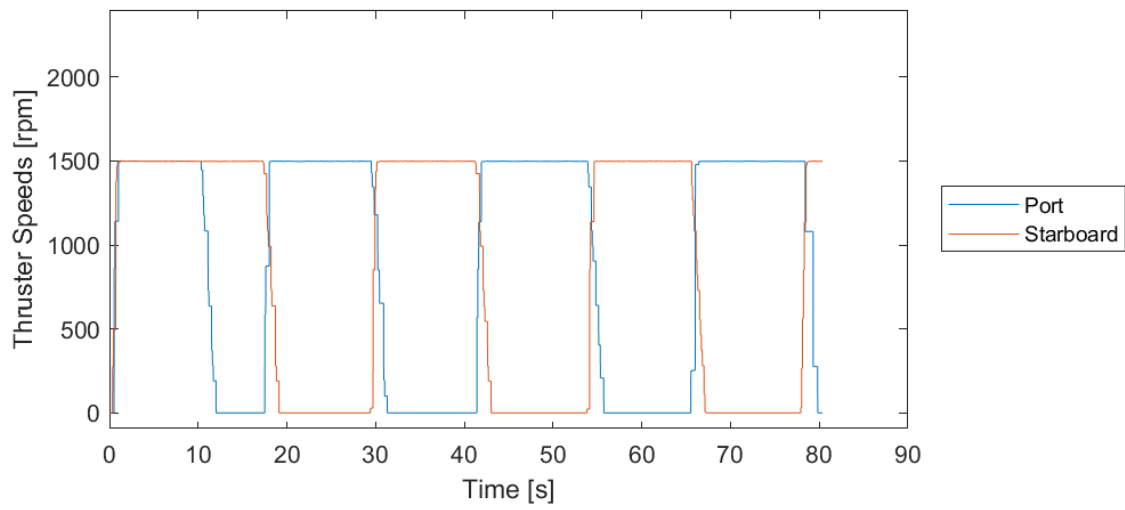
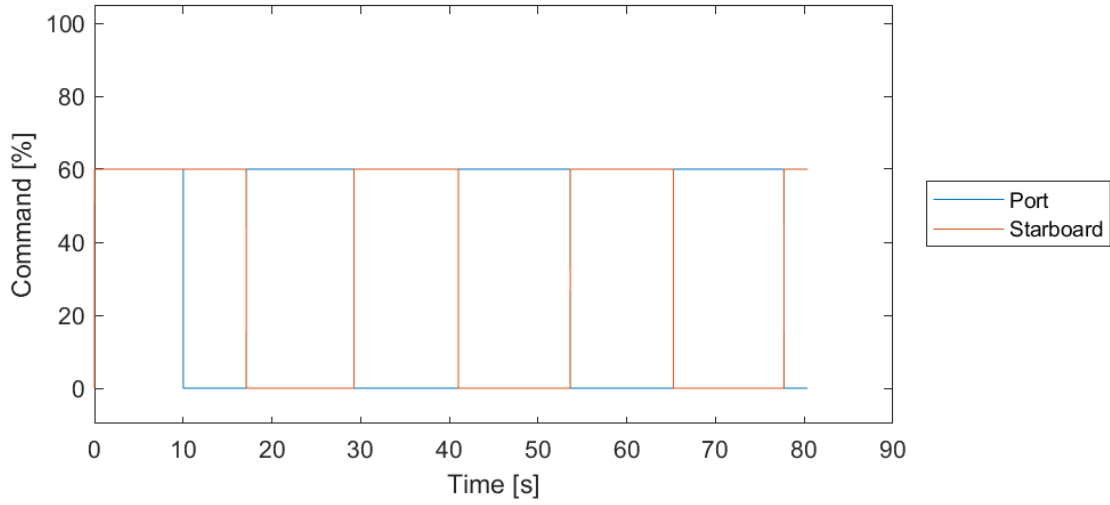


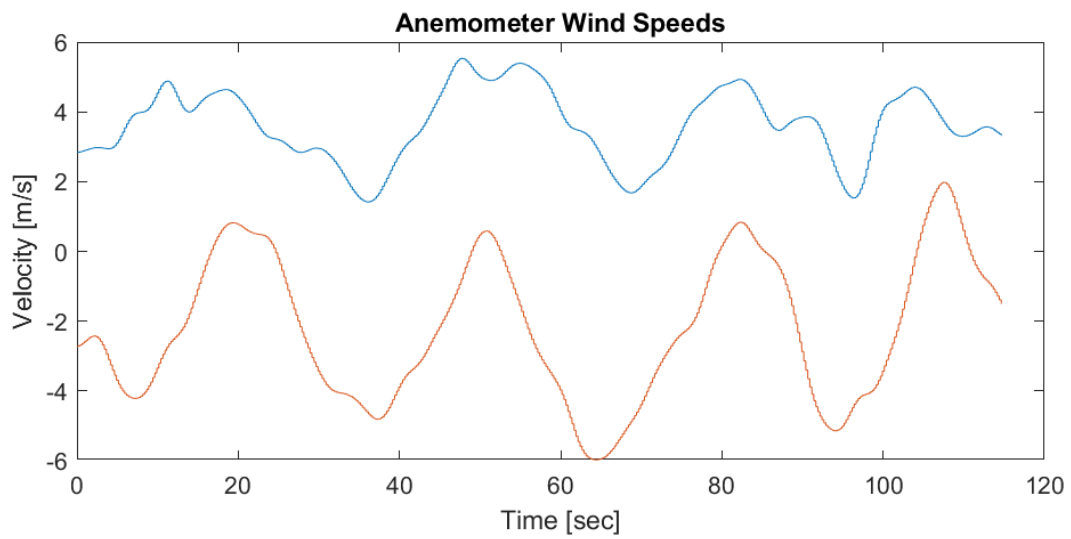
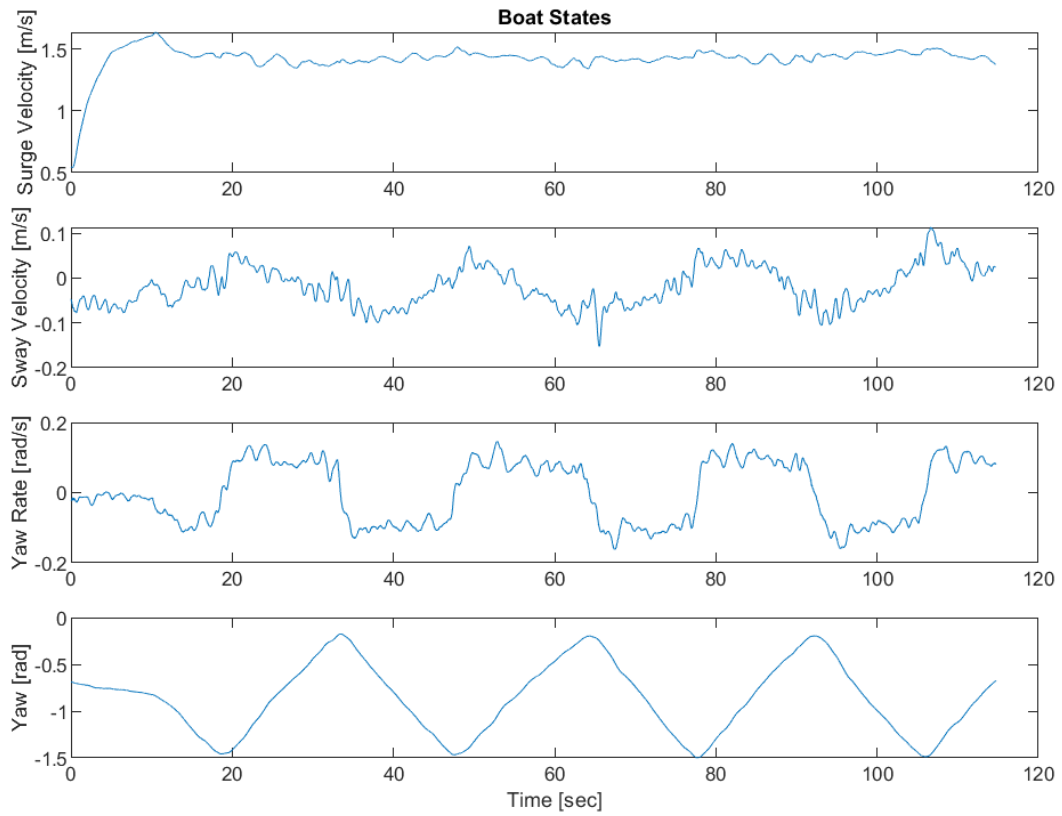
1.5.16. Zig-Zag Set 16

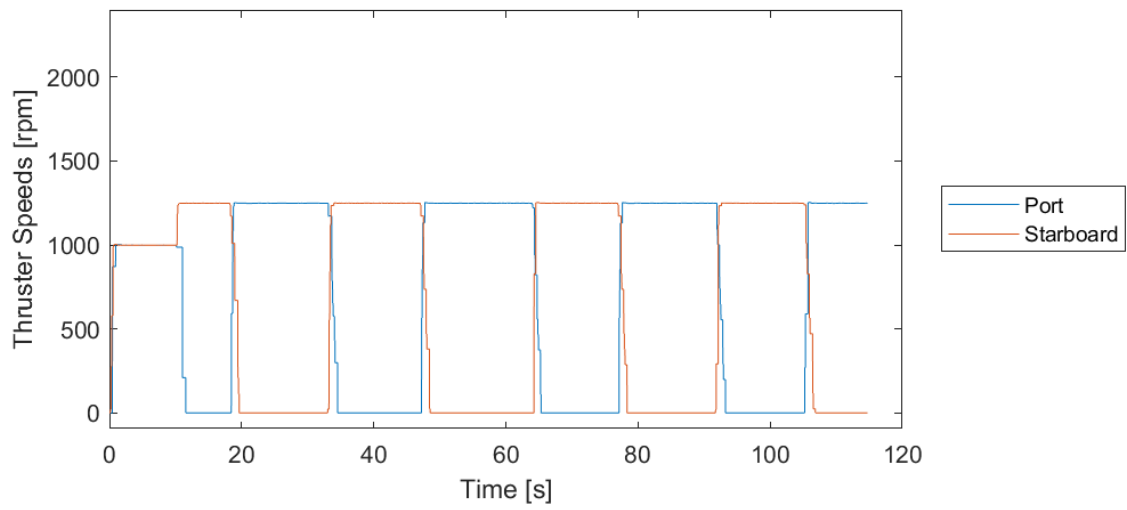
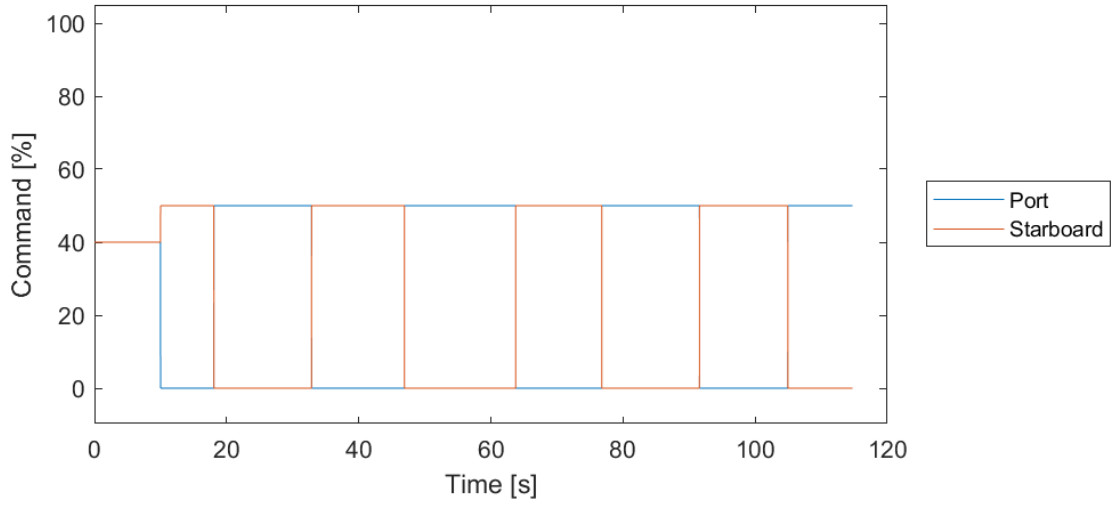




1.5.17. **Zig-Zag Set 17**

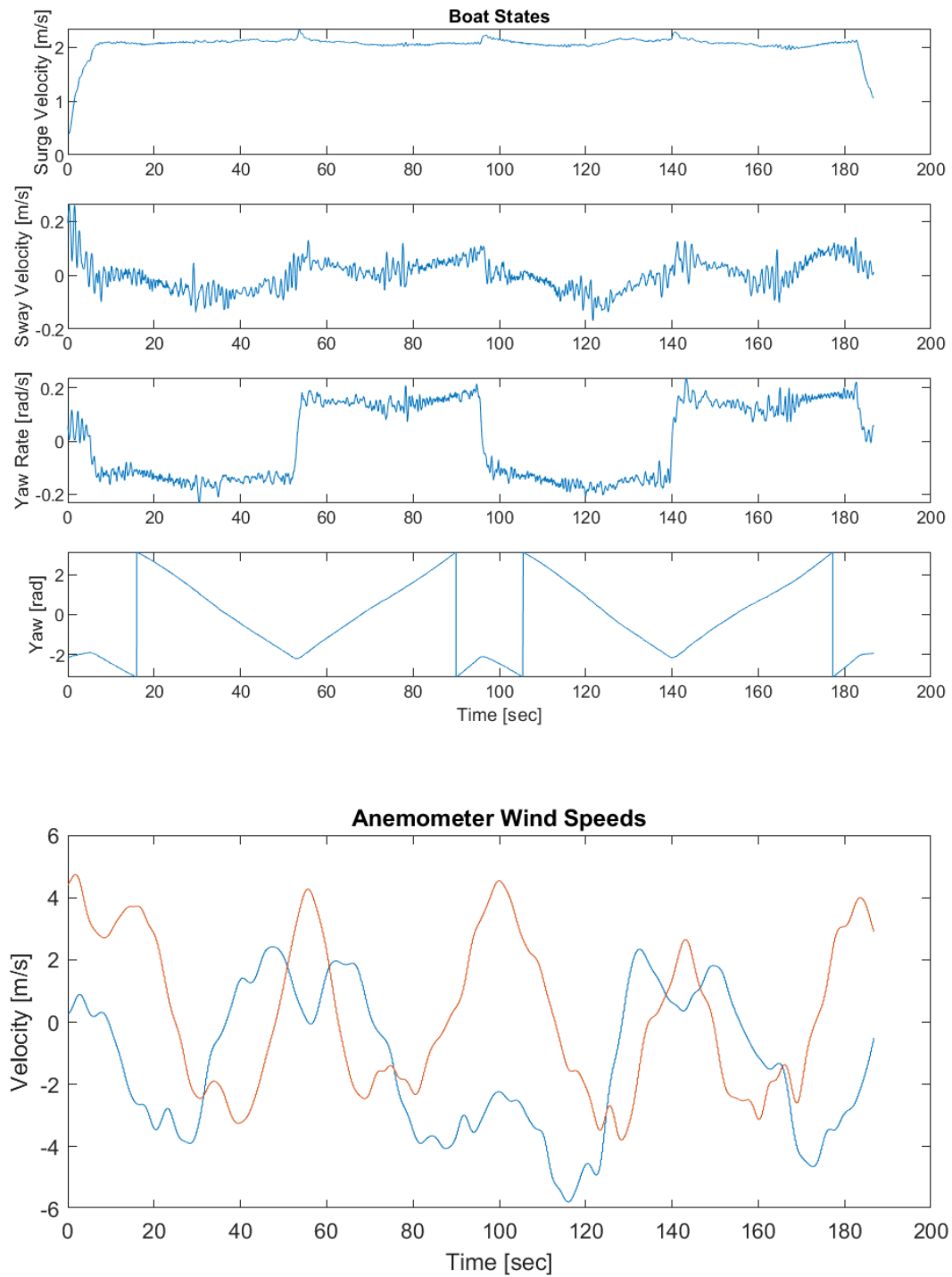


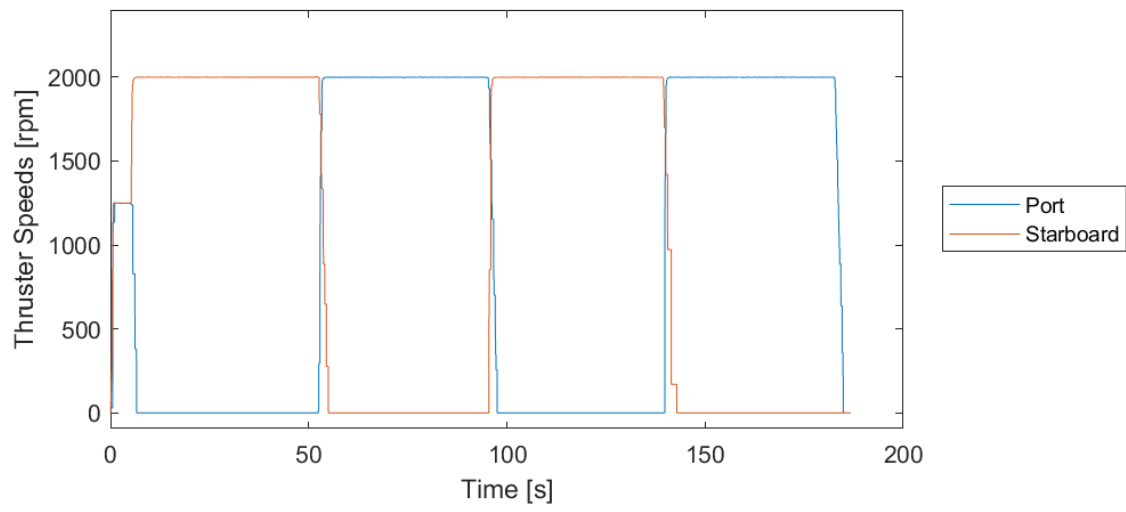
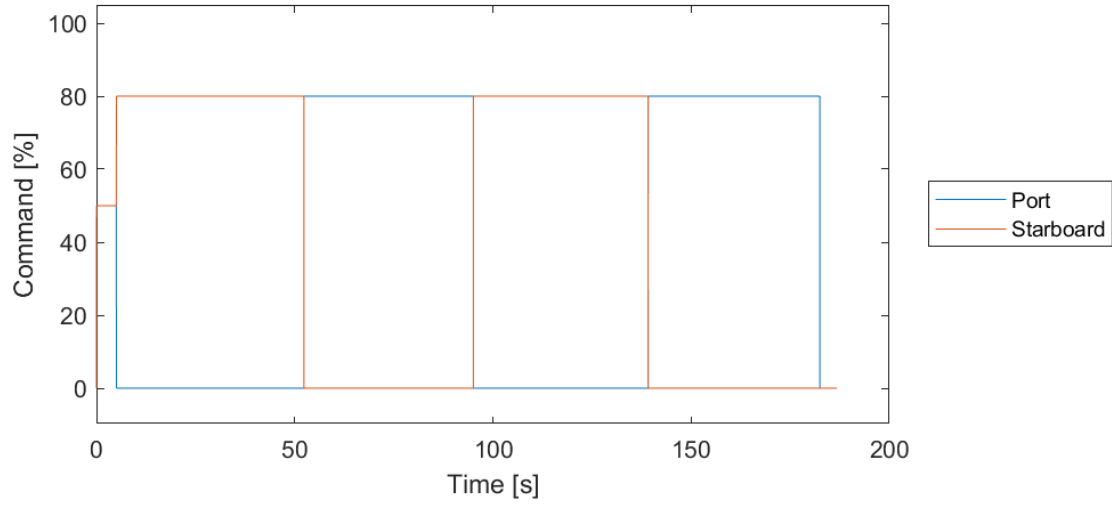
1.5.18. **Zig-Zag Set 18**



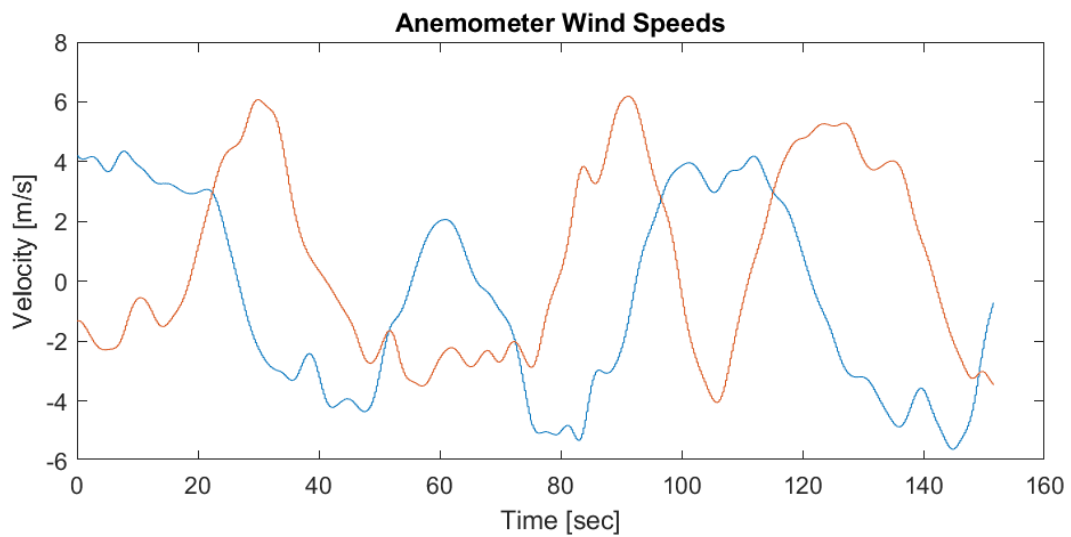
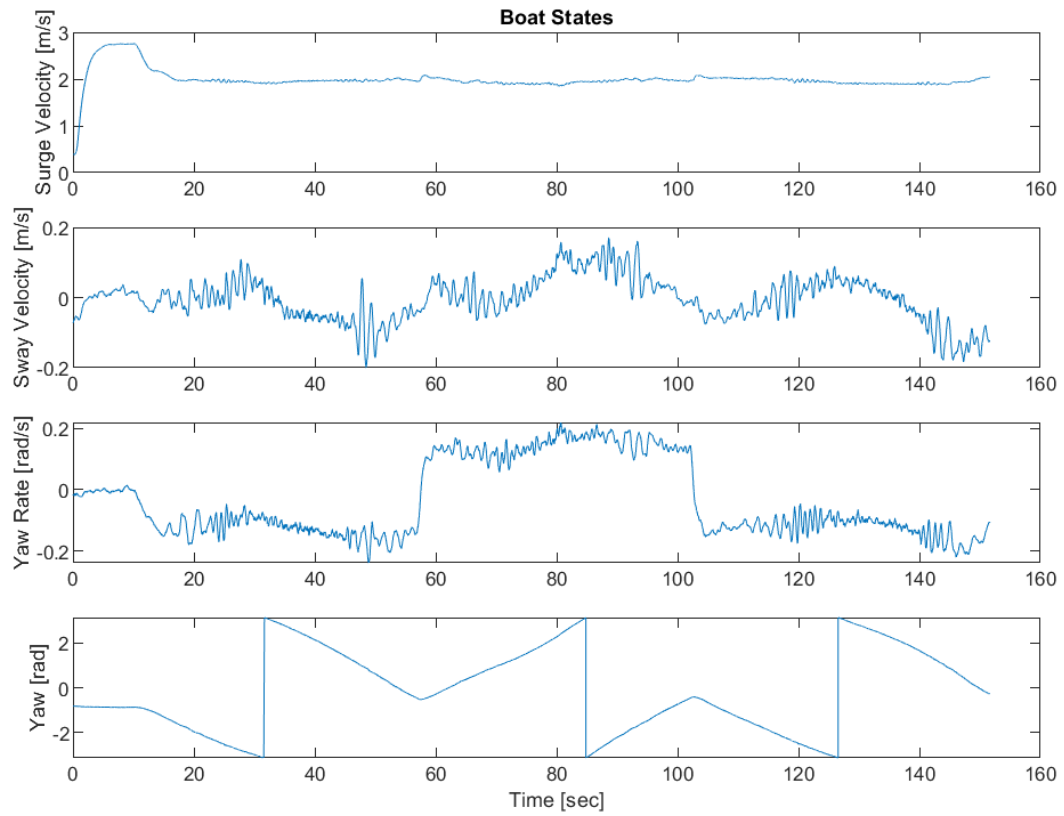
1.6. Figure Eight Data Sets

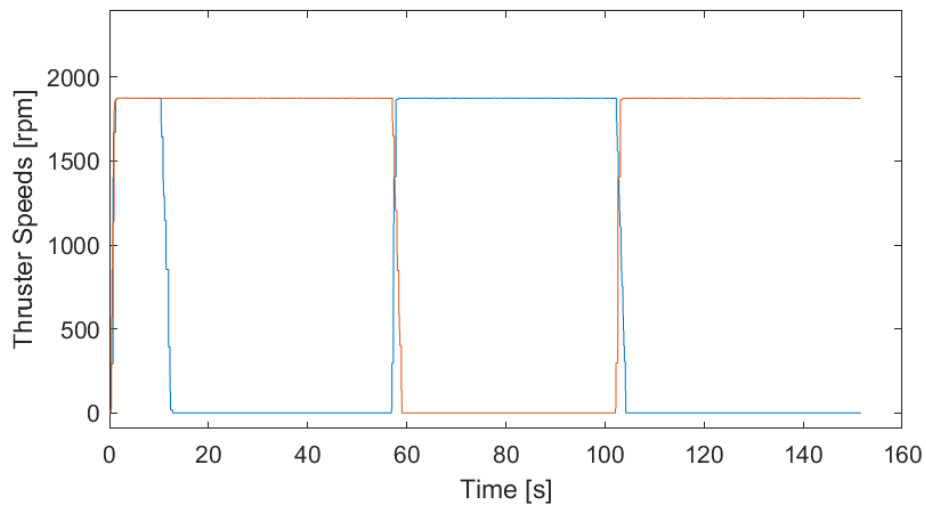
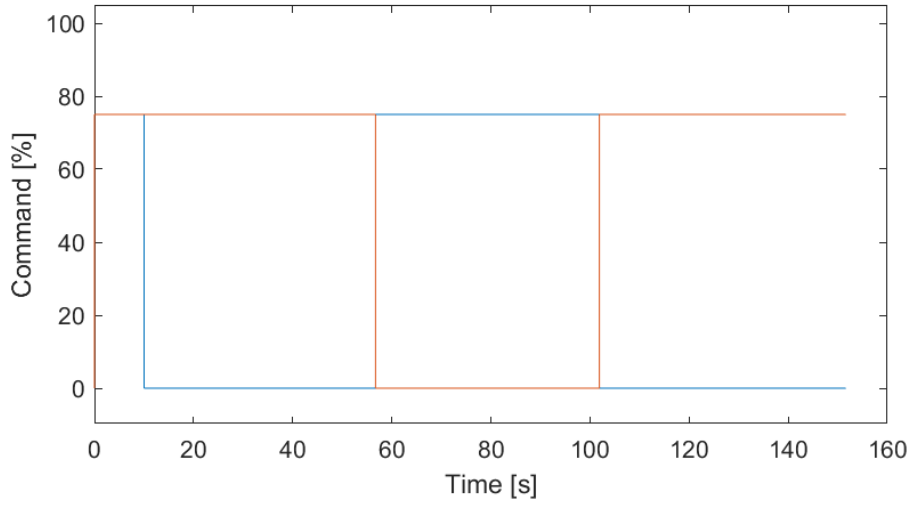
1.6.1. Figure Eight Set 1





1.6.2. Figure Eight Set 2





Genetic Algorithm Code

2.1. GA Optimizer

This is the optimizer code that takes in multiple “.mat” logs and optimizes them serially.

The GA settings are set in advance and the population and fitness scores carry over to initialize the next log.

```

clc
clear
rng default;
warning('off','all')
distcomp.feature('LocalUseMpiexec', false); % Allows parallel in 2015b
format shortG

pause(1)

wind_dir      = dir('Logs\Optimize Wind\*.mat');
opt_dir       = dir('Logs\Optimize\*.mat');
parallel      = false;
validate      = false;
eval_hrs      = 6;

max_gen       = 500;
max_stall_gen = 75;
pop_size      = 50;

start_opt_time = now;
start_tot_time = tic;
log_time_str   = datestr(start_opt_time, 'mm_dd_yy__HH_MM_SS');

if parallel
    try
        myCluster = parcluster('local');
        parpool(myCluster.NumWorkers);
    catch
    end
else
    poolobj = gcp('nocreate');
    delete(poolobj);
end

stop_weight = [1, 1, 3, 0, 0, 1.5]';
zz_weights  = [1, 1, 3, 0, 0, 2]';
turn_weight = [1, 1, 3, 0, 0, 3]';
fe_weight   = [1, 1, 3, 0, 0, 3]';
wind_weight = [1, 1, 3, 0.5, 0.5, 3]';
spin_weight = [1, 1, 3, 0, 0, 3]';

% Search for general keywords in log names
log_type_names = {'stop'; 'zig'; 'turn'; 'figure'; 'wind'; 'spin'};
log_type_weights= {stop_weight; zz_weights; turn_weight; fe_weight;...

```

```

    wind_weight; spin_weight});

log_data      = cell(size(wind_dir)+size(opt_dir));
t             = cell(size(wind_dir)+size(opt_dir));
weights       = cell(size(wind_dir)+size(opt_dir));
w_s = length(wind_dir);

for i = 1:length(wind_dir)
    load(fullfile('Logs\Optimize Wind\'', wind_dir(i).name))
    log_data{i} = [state current input wind];
    t{i}        = time;
    for j = 1:length(log_type_names)
        if(~isempty(strfind(wind_dir(i).name, log_type_names(j))))
            weights{i} = log_type_weights{j};
            break;
        end
    end
end

for i = 1:length(opt_dir)
    load(fullfile('Logs\Optimize\'', opt_dir(i).name))
    log_data{i+w_s} = [state current input wind];
    t{i+w_s}        = time;
    for j = 1:length(log_type_names)
        if(~isempty(strfind(opt_dir(i).name, log_type_names(j))))
            weights{i+w_s} = log_type_weights{j};
            break;
        end
    end
end

% Randomizes order of logs
rand_logs = randperm(length(opt_dir));

log_names = [wind_dir; opt_dir(rand_logs)];
log_data  = [log_data(1:length(wind_dir))
log_data(length(wind_dir)+rand_logs)];
t         = [t(1:length(wind_dir)) t(length(wind_dir)+rand_logs)];
weights   = [weights(1:length(wind_dir))
weights(length(wind_dir)+rand_logs)];
eval_hrs  = eval_hrs * ones(length(log_data), 1);

fprintf('Logs being optimized in order:\n')
fprintf('*****\n\n')
for i = 1:length(opt_dir)+length(wind_dir)
    fprintf('%s\n', log_names(i).name)
end
fprintf('\n*****\n\n')
%% Model Settings

% Rigid-Body Parameters
mass = 348.390;
Izz  = 525.390;

A_fw = 81.451;
A_lw = 702.89;

```

```

L_aa = 4.55;

% Hydrodynamic Lower Limits
%      Xudot   Xu   Xuu Xvdot   Xv   Xvv Xrdot   Xr   Xrr
X_LB   = [ 450   0   50   0   0   -100   0   0   -130];
%      Yudot   Yu   Yuu Yvdot   Yv   Yvv Yrdot   Yr   Yrr
Y_LB   = [ 0   0   0  1000  175  500   0   0  -2500];
%      Nudot   Nu   Nuu Nvdot   Nv   Nvv Nrdot   Nr   Nrr
N_LB   = [ 0   0  -700   0   0  -2000  600   0  1000];
%      Yvr
%      Yrv
Y_C_LB = [ -2000 -2000];
%      Nvr
%      Nrv
N_C_LB = [ -3000 -3000];
%      X CG   Y CG
R_LB   = [-0.15 0.23];
%      vc   ac
C_LB   = [ 0   0];
%      Cx   Cy   Cz
W_LB   = [0.001 0.001 0.0001];

% Hydrodynamic Lower Limits
%      Xudot   Xu   Xuu Xvdot   Xv   Xvv Xrdot   Xr   Xrr
X_UB   = [ 550  100  175  500  100  100  200  130  130];
%      Yudot   Yu   Yuu Yvdot   Yv   Yvv Yrdot   Yr   Yrr
Y_UB   = [ 0   0   0  2500  1800  2000  2000  3000  2500];
%      Nudot   Nu   Nuu Nvdot   Nv   Nvv Nrdot   Nr   Nrr
N_UB   = [ 500  700  700  3000  5000  2000  3500  1500  3000];
%      Yvr
%      Yrv
Y_C_UB = [ 0  1000];
%      Nvr
%      Nrv
N_C_UB = [ 1000 0];
%      X CG   Y CG
R_UB   = [-0.1 0.28];
%      vc   ac
C_UB   = [ 0   0];
%      Cx   Cy   Cz
W_UB   = [ 0.15 0.15 0.01];

%% Optimizer Settings

% weights = [u_w; v_w; r_w; N_w; E_w; Y_w];
rb_coeffs = [mass, Izz, A_fw, A_lw, L_aa];

LB      = [X_LB, Y_LB, Y_C_LB, N_LB, N_C_LB, R_LB, C_LB, W_LB];
UB      = [X_UB, Y_UB, Y_C_UB, N_UB, N_C_UB, R_UB, C_UB, W_UB];

pop     = [];
scores  = [];

%% Optimizer
save_folder = ['Results\' log_time_str ' - Boat Optimizer Results\'];
mkdir(save_folder)

pause(3)

```

```

diary_log = [save_folder 'Optimization Evaluation Log - ' log_time_str
'.txt'];
diary(diary_log)

curr_log = cell(1,1);
curr_time = cell(1,1);
curr_weight = cell(1,1);

fprintf('Optimization Start! at %s\n', log_time_str)
fprintf('\n')

for InProgress = 1:length(opt_dir)
    curr_log = log_data(InProgress);
    curr_time = t(InProgress);
    curr_weight= weights{InProgress};

    f = @(coeffs)boat_error_parallel(coeffs, curr_time, curr_time, curr_log,
rb_coeffs, curr_weight, true);

    eval_time = eval_hrs(InProgress) * 3600; % Time to search space in
seconds

    options = optimoptions('ga', 'Display', 'iter',...
'MaxTime', eval_time,...
'MaxGenerations', max_gen,...
'MaxStallGenerations', max_stall_gen,...
'InitialPopulation', pop,...
'InitialScores', scores,...
'PopulationSize', pop_size,...
'PlotFcn', {@gaplotbestf, @gaplotbestindiv, @gaplotrange},...
'UseParallel', parallel, 'UseVectorized', false);

    log_start_time = now;

    fprintf('Optimization Start for %s at %s\n',
opt_dir(InProgress).name(1:end-4), datestr(log_start_time, 'mmm dd, yyyy
HH:MM:SS.FFF AM'))
    start_time = tic;

    % Genetic Algorithm Search
    try
        [coeffs, fval, exitflag, output, pop, scores] =
ga(f,length(LB), [], [], [], [], LB,UB, [], options);
        fval
        exitflag
        output
    catch e
        fprintf('GA Optimizer failed:\n\n')
        disp(e)
        fprintf('\n')
    end

    total_time = toc(start_time);
    end_time = now;
    fprintf('Optimization Ended for %s at %s\n', datestr(end_time, 'mmm dd,
yyyy HH:MM:SS.FFF AM'))

```

```

fprintf('\n')

hrs = floor(total_time / 3600);
total_time_c = total_time - hrs * 3600;
min = floor(total_time_c / 60);
sec = total_time_c - min * 60;

LB      = [X_LB, Y_LB, Y_C_LB, N_LB, N_C_LB, R_LB, C_LB, W_LB];

x_l = length([X_LB]);
y_l = length([Y_LB Y_C_LB]);
n_l = length([N_LB N_C_LB]);
r_l = length([R_LB]);
c_l = length([C_LB]);
w_l = length([W_LB]);

rb_coeffs

if ~isempty(coeffs)

    X_coeffs = coeffs(1:x_l)
    Y_coeffs = coeffs(1+x_l:x_l+y_l)
    N_coeffs = coeffs(1+x_l+y_l:x_l+y_l+n_l)
    R_coeffs = coeffs(1+x_l+y_l+n_l:x_l+y_l+n_l+r_l)
    C_coeffs = coeffs(1+x_l+y_l+n_l+r_l:x_l+y_l+n_l+r_l+c_l)
    W_coeffs = coeffs(1+x_l+y_l+n_l+r_l+c_l:x_l+y_l+n_l+r_l+c_l+w_l)

else
    fprintf('\nNo coefficients we generated! >:(\n\n')
end

coeffs

fprintf('Optimization Time for %s: %d hrs %d min %0.3f sec\n',
opt_dir(InProgress).name(1:end-4), hrs, min, sec)
fprintf('\n')
fprintf('*****\n\n')
save_filename = [save_folder num2str(InProgress) '_' log_time_str, '- '
opt_dir(InProgress).name(1:end-4) ' - Boat Optimizer Results'];
save(save_filename)

saveas(gcf, [save_filename, ' GA Plot'])
close all
end

%% Results

total_opt_time = toc(start_tot_time);

hrs_tot      = floor(total_opt_time / 3600);
total_opt_time = total_opt_time - hrs_tot * 3600;
min_tot      = floor(total_opt_time / 60);
sec_tot      = total_opt_time - min_tot * 60;

fprintf('\n')

```

```
fprintf('Total Optimization Time: %d hrs %d min %0.3f sec\n', hrs_tot,  
min_tot, sec_tot)  
  
save_filename = [save_folder log_time_str, ' - Boat Optimizer Results'];  
save(save_filename)  
  
diary off
```

2.2. Genetic Algorithm Error Function

The GA error function that reports the weighted mean squared error of all the states of the dynamic system.

```
function error = boat_error_parallel(coeffs, t, time, log_data, rb_coeffs,
weights, global_dist)

warning('off','all')
states_out = cell(length(time), 1);
prev_err = 0;
valid_case = zeros(1, length(states_out));

for i = 1:length(states_out)

    input = log_data{i}(:, 9:12);
    state = log_data{i}(:, 1:6);
    current = log_data{i}(:, 7:8);
    wind = log_data{i}(:, 13:14);
    t_p = t{i};
    time_p = time{i};

    f = @(t,s) MinionBoatModel(t, time_p, s, current, wind, input, rb_coeffs,
coeffs, global_dist);

    [~, states_out{i}] = ode45(f, t_p, state(1,:));

end

for j = 1:length(states_out)
    if (size(log_data{j}(:, 1:6)) == size(states_out{j}))

        curr_err = sum( ((log_data{j}(:, 1:5) - states_out{j}(:, 1:5)).^2) ,
1) * weights(1:5);
        curr_err = curr_err + sum( (wrapToPi(log_data{j}(:, 6) -
states_out{j}(:, 6)).^2) .* weights(6) , 1);

        prev_err = prev_err + curr_err;

        valid_case(j) = 1;
    else
        valid_case(j) = 0;
    end
end

if sum(valid_case) > 0
    error = (prev_err / sum(valid_case));
else
    error = 100000001;
end
if (isnan(error))
    error=100000002;
end
```

2.3. Minion Boat Model

The system model function used in this thesis work. This model is based on the 2nd-order maneuvering Fossen model.

```
function ds = MinionBoatModel(t, time, state,...
                             current, wind, input,...
                             rb_coeffs, coeffs, global_dist)

%% Settings
%-----
% Set Coeficients
%-----
% Surge Force
Xud = coeffs(1);
Xu  = coeffs(2);
Xuu = coeffs(3);
Xvd = coeffs(4);
Xv  = coeffs(5);
Xvv = coeffs(6);
Xrd = coeffs(7);
Xr  = coeffs(8);
Xrr = coeffs(9);
% Sway Force
Yud = coeffs(10);
Yu  = coeffs(11);
Yuu = coeffs(12);
Yvd = coeffs(13);
Yv  = coeffs(14);
Yvv = coeffs(15);
Yrd = coeffs(16);
Yr  = coeffs(17);
Yrr = coeffs(18);
Yvr = coeffs(19);
Yrv = coeffs(20);
% Yaw Moments
Nud = coeffs(21);
Nu  = coeffs(22);
Nuu = coeffs(23);
Nvd = coeffs(24);
Nv  = coeffs(25);
Nvv = coeffs(26);
Nrd = coeffs(27);
Nr  = coeffs(28);
Nrr = coeffs(29);
Nvr = coeffs(30);
Nrv = coeffs(31);
% Rigid-Body
m    = rb_coeffs(1);
Izz  = rb_coeffs(2);
A_fw = rb_coeffs(3);
A_lw = rb_coeffs(4);
L_aa = rb_coeffs(5);

x_cg = coeffs(32);
y_cg = coeffs(33);
% Current
```



```

c_vel = coeffs(34);
c_ang = wrapToPi(coeffs(35));
% Wind
cx_w = coeffs(36);
cy_w = coeffs(37);
cz_w = coeffs(38);

%-----
% Set States
%-----
u      = state(1);
v      = state(2);
r      = state(3);
N      = state(4);
E      = state(5);
Y      = wrapToPi(state(6));

%-----
% Model Inputs
%-----
% Disturbances
nu      = [u ; v ; r];

% Current
if global_dist
    cN = c_vel*cos(c_ang);
    cE = c_vel*sin(c_ang);

    cN = interp1(time, current(:, 1), t);
    cE = interp1(time, current(:, 2), t);

    cx = cN*cos(Y) - cE*sin(Y);
    cy = cN*sin(Y) + cE*cos(Y);
else
    cx = interp1(time, current(:, 1), t);
    cy = interp1(time, current(:, 2), t);
end

ur      = u - cx;
vr      = v - cy;
nur      = [ur; vr; r];

% Wind
u_rw = interp1(time, wind(:, 1), t);
v_rw = interp1(time, wind(:, 2), t);

T_w = Wind_Model(u_rw, v_rw, cx_w, cy_w, cz_w, A_fw, A_lw, L_aa);

% Motor
T_tf = [1 0 1 0; 0 1 0 1; 1-y_cg -1.71-x_cg -1-y_cg -1.71-x_cg];
T      = (T_tf*interp1(time, input, t)');

%% Model Dynamics

```

```

%-----
% Mass Matrix
%-----
M_RB = [      m,      0, -m*y_cg;...
         0,      m,  m*x_cg;...
        -m*y_cg,  m*x_cg,  Izz];

M_AM = [  Xud,  Xvd,  Xrd;...
         Yud,  Yvd,  Yrd;...
         Nud,  Nvd,  Nrd];

M      = M_RB + M_AM;

%-----
% Coriolis Effect
%-----
C_RB = [      0,      0, -m*(x_cg*r + v);...
         0,      0, -m*(y_cg*r - u);...
        m*(x_cg*r + v),  m*(y_cg*r - u),  0];

C_AM = [      0,      0,  Yud*ur+Yvd*vr+Yrd*r;...
         0,      0, -Xud*ur-Xvd*vr-Xrd*r;...
        -Yud*ur-Yvd*vr-Yrd*r,  Xud*ur+Xvd*vr+Xrd*r,  0];

%-----
% Drag Terms
%-----
D_L  = [  Xu,      Xv,      Xr;...
         Yu,      Yv,      Yr;...
         Nu,      Nv,      Nr];

D_NL = [Xuu*abs(ur),      Xvv*abs(vr),      Xrr*abs(r);...
        Yuu*abs(ur),  Yvv*abs(vr) + Yrv*abs(r),  Yrr*abs(r) + Yvr*abs(vr);...
        Nuu*abs(ur),  Nvv*abs(vr) + Nrv*abs(r),  Nrr*abs(r) + Nvr*abs(vr)];

D      = D_L + D_NL;

%-----
% Accelerations
%-----
ds     = [M\((T + T_w) - C_RB*nu - C_AM*nur - D*nur);...
         u*cos(Y) - v*sin(Y)      ;...
         u*sin(Y) + v*cos(Y)      ;...
         r                          ];
```

end