

University of Alberta

Formation Control for Autonomous Marine Vehicles

by

Christopher John Van Kleeck

A thesis submitted to the Faculty of Graduate Studies and Research
in partial fulfillment of the requirements for the degree of

Master of Science

Department of Mechanical Engineering

©Christopher John Van Kleeck

Fall 2009

Edmonton, Alberta

Permission is hereby granted to the University of Alberta Libraries to reproduce single copies of this thesis and to lend or sell such copies for private, scholarly or scientific research purposes only. Where the thesis is converted to, or otherwise made available in digital form, the University of Alberta will advise potential users of the thesis of these terms.

The author reserves all other publication and other rights in association with the copyright in the thesis and, except as herein before provided, neither the thesis nor any substantial portion thereof may be printed or otherwise reproduced in any material form whatsoever without the author's prior written permission.

EXAMINING COMMITTEE

Dr. Fahimi, Mechanical Engineering

Dr. C. R. Koch, Mechanical Engineering

Dr. Behzadipour, Mechanical Engineering

Dr. Dinavahi, Electrical and Computer Engineering

ABSTRACT

The development, implementation, and testing of a leader-follower based robust nonlinear formation controller is discussed in this thesis. This controller uses sliding mode control on the length and angle between the leader and follower vessels to produce the desired formation. A boat model, assuming planar motion (three degrees of freedom), is used as the bases for the controller.

Open loop testing is performed to determine parameter values to match the simulation model to the physical one and, upon tuning of the controller to match, closed loop testing of the controller with a virtual leader is also performed. From these tests it is found that the controller is unstable, thus improvements to the controller, through changes made to the model and to the parameter identification process, are undertaken. Simulations comparing the initial and updated models of the vehicle to open loop data show an improvement in the new model.

ACKNOWLEDGEMENTS

This thesis could not have been done without the support and guidance received from Dr. Farbod Fahimi, for this I am supremely grateful. I am also thankful for the camaraderie and assistance of my lab mates: Mehdi Saffarian, Mojtaba Azadi, Siavash Rezazadeh, Sepehr Pourezai, and lab alumni Charles Yang. I am also grateful for the assistance of summer students Matthew Bourassa and Michael Dawson with the set up of the hardware.

CONTENTS

1	Introduction and Literature Review	1
2	Simplified Simulation Design	5
2.1	Dynamic Boat Plant Model	6
2.2	Simulation of Plant Model	9
2.3	l - ψ Controller Design	18
2.3.1	Input-Output Relations	19
2.3.2	Sliding Mode Control Law	23
2.3.3	Simulink Implementation	26
2.4	Overall System	27
2.4.1	Lead Boats	27
2.4.2	Straight Line Trajectory	29
2.4.3	Circular Trajectory	30
2.4.4	Sinusoidal Trajectory	32
2.4.5	Complete System Assembly	32
2.5	Simulation and Discussion	35
2.5.1	Simulation of the closed-loop system	35
2.5.2	Discussion	38
3	Applicable Design	41
3.1	Dynamic Model: Plant Equations Derivation	41

3.2	L-Psi Controller Design: Controller Derivation	44
3.3	Simulation and Discussion	47
3.3.1	Lead Boat Paths	48
3.3.2	Simulation results	61
3.3.3	Discussion	69
4	Experimental Setup	70
4.1	Sensor	70
4.1.1	Setup	71
4.1.2	Configuration and Calibration	71
4.2	Drive Train Control: SSC Configuration and Setup	72
4.3	Onboard Computer	73
4.4	Controller Box	75
4.5	Boat Hull Reconfiguration	75
4.5.1	Design	75
4.5.2	Fabrication	78
5	Experimental Results - Simple Model	82
5.1	Integration of subsystems and real-time control program	82
5.2	Differential Coordinate System	84
5.3	Experimental Setup	85
5.4	Closed Loop Setup	90
5.5	Experiment Results and Discussion	93
6	Experimental Results - Enhanced Model	101
6.1	Update of Boat Model	101
6.2	Update of Propeller Thrust Relation	102
6.3	Parameter Identification with Updated Model	103

6.4	Comparison of Models	108
6.5	Robustness of Enhanced Controller	113
7	Conclusions and Future Work	116
7.1	Contributions	116
7.2	Conclusions	116
7.3	Future Work	119
	Bibliography	121
A	Embedded Source Code	124
B	Controller Box: Construction Procedure	126
C	Hardware and Software Lists	129

LIST OF TABLES

4.1	Weights of Components	76
6.1	Grey-box Identification Parameter Constraints	105
6.2	Grey-box Identification Parameter Constraints Continued	105
6.3	Parameter Results for Grey-box Identification with Data Set # 41 . .	106
6.4	Error Comparison across Data sets	107

LIST OF FIGURES

2.1	Boat Diagram	7
2.2	Boat Plant Interface	9
2.3	Boat Plant	11
2.4	Straight Line Boat Response (a) Position response; (b) Velocity response.	12
2.5	Straight Line Boat Response: Boat trajectory with orientation.	13
2.6	Circular Boat Response (a) Position response; (b) Velocity response. .	14
2.7	Circular Boat Response: Boat trajectory with orientation.	15
2.8	Sinusoidal Boat Response (a) Position response; (b) Velocity response.	16
2.9	Sinusoidal Boat Response: Boat trajectory with orientation.	17
2.10	System States	19
2.11	Controller Diagram	27
2.12	Formation Parameters (controller outputs) Calculations	28
2.13	Lead Boat: Straight trajectory	29
2.14	Lead Boat: Circular trajectory	31
2.15	Lead Boat: Sinusoidal trajectory	33
2.16	Overall System Diagram	34
2.17	System Response: Straight lead path Follower begins at origin and lead boat starts in (x, y) position of $(0,10)$	36
2.18	System Response: Circular lead path Lead boat starts at $(0,10)$, fol- lower starts at origin	36

2.19	System Response: Sinusoidal lead path Lead boat begins at position (0,10), follower starts at origin	37
2.20	System Response: Straight line case system outputs l_{12} and ψ_{12}	39
2.21	System Response: Circular case system outputs l_{12} and ψ_{12}	40
2.22	System Response: Sinusoidal case system outputs l_{12} and ψ_{12}	40
3.1	Boat Loading Diagram	43
3.2	New Boat Plant Diagram	44
3.3	Straight Lead Boat: (a) x component; (b) y component.	50
3.4	Straight Lead Boat: (a) θ component; (b) Boat trajectory with orientation.	51
3.5	Circular Lead Boat: (a) Left: x component; (b) y component.	53
3.6	Circular Lead Boat: (a) θ component; (b) Boat trajectory with orientation.	54
3.7	Example of curve with starting and ending times and positions	56
3.8	Zigzag Lead Boat: (a) x component; (b) y component.	59
3.9	Zigzag Lead Boat: (a) θ component; (b) Boat trajectory with orientation.	60
3.10	Case 1: Simulation of straight line path (a) Boat trajectories with orientations; (b) Follower inputs.	62
3.11	Case 1: Simulation of straight line path (a) System outputs; (b) Derivatives.	63
3.12	Case 2: Simulation of circular path (a) Boat trajectories with orientations; (b) Follower inputs.	65
3.13	Case 2: Simulation of circular path (a) System outputs; (b) Derivatives.	66
3.14	Case 3: Simulation of zigzag shaped path (a) Boat trajectories with orientations; (b) Follower inputs.	67

3.15 Case 3: Simulation of zigzag shaped path (a) System outputs; (b) Derivatives.	68
4.1 Rough Dimensions of Boat (in centimeters)	76
4.2 The Boat Mount Platform	77
4.3 Diagram of Cutting lines for Shaping pontoons	78
4.4 The pontoons and connecting spars	79
4.5 The Boat Mount Connected to pontoons via spars	80
4.6 Close view of clamps used to fasten Boat Mount to the hull of the boat	81
4.7 The Complete Assembly on the Boat	81
5.1 Implementation Interface	84
5.2 Inputs for Identification Tests	86
5.3 Open Loop Boat Trajectory from Test 2	87
5.4 Recorded and Simulated Velocities for Test 2	88
5.5 Recorded and Simulated Velocities for Test 1	90
5.6 Recorded and Simulated Velocities for Test 3	91
5.7 Recorded and Simulated Velocities for Test 4	92
5.8 Closed Loop Boat Response: Trajectory	93
5.9 Closed Loop Controller Outputs; with saturation levels of 140 rps and ± 45 degrees	94
5.10 Closed Loop Boat Response: Velocities	95
5.11 Simulation of Boat Response: Trajectory	96
5.12 Simulation of Boat Response: Velocities	97
5.13 Closed Loop Boat Response: Trajectory	98
5.14 Closed Loop Boat Response: Velocities	99
5.15 Closed Loop Boat Response: Controller Inputs	100

6.1	Comparison of Boat velocities for old model vs. new model vs. actual boat; with no simulated disturbances, from Data Set 43	109
6.2	Comparison of Boat velocities for old model vs. new model vs. actual boat; with disturbances included, from Data Set 42	110
6.3	Comparison of Boat velocities for old model vs. new model vs. actual boat, Data Set 42	111
6.4	Comparison of Boat velocities for old model vs. new model vs. actual boat, Data Set 44	112
6.5	Sum of velocity errors for varying % uncertainty in parameter values .	114
6.6	System outputs $l - \psi$ for varying % uncertainty in parameter values .	115

NOMENCLATURE

Acronyms

AHPRS Altitude Heading Position Reference System

CG Center of Gravity

DOF Degrees of Freedom

DOS Disk Operating System

FTP File Transfer Protocol

GPS Global Positioning System

HDD Hard Disk Drive

IP Internet Protocol

Li-Po Lithium-Polymer

PC Personal Computer

PID Proportional-Integral-Derivative

RC Radio Controlled

rpm Rotations per Minute

rpm	Rotations per Second
SSC	Servo Switch/Controller
TCP	Transmission Control Protocol
UDP	User Datagram Protocol

Symbols:

Vectors and Matrices

\mathbf{b}	Controller Matrix
η	Controller Offsets
\mathbf{f}	Dynamics of Vessel
\mathbf{k}	Controller Gains
\mathbf{m}	Mass and Moment of Inertia
ϕ	Boundary Layer Thicknesses
\mathbf{q}	Boat State (position and velocity)
\mathbf{s}	Scalar, Sliding Surface
\mathbf{s}_r	Desired Sliding Surface
\mathbf{v}_p	Velocity of Control Point
\mathbf{u}	Inputs, Driving Forces and Torques
\mathbf{z}	System Outputs

Majuscules

D	Diameter of Propellers [m]
J_0	Advance Coefficient
K_T	Thrust Coefficient
L	Distance from propellers to CG [m]
N	Driving Moment about Yaw [Nm]
T_h	Simplified Thrust Term [N]
V_a	Advance Speed [m/s]
X	Driving Force in Surge Direction [N]
Y	Driving Force in Sway Direction [N]

Minuscules

d	Distance from CG to Control Point [m]
d_{11}, d_{22}	Damping Coefficient (linear) [$\frac{kg}{s}$]
d_{23}, d_{32}	Damping Coupling Terms [$\frac{kg \cdot m}{s}$]
d_{33}	Damping Coefficient (rotational) [$\frac{kg \cdot m^2}{s}$]
l_{12}	Distance between leader CG and follower control point [m]
m_{11}, m_{22}	Mass (including hydrodynamic added mass) [kg]
m_{23}, m_{32}	Mass Coupling Terms [$kg \cdot m$]
m_{33}	Moment of Inertia [$kg \cdot m^2$]
n	Engine Speed [rps]
r	Yaw rate [$\frac{radian}{s}$]
t	Time [s]
u	Surge Speed [$\frac{m}{s}$]
v	Sway Speed [$\frac{m}{s}$]
x	Position (North Positive) [m]
y	Position (East Positive) [m]

Greek Minuscles

α	Rudder Angle [rad]
η	Controller Offset
λ	Sliding Mode Weighting Factor [1/s]
ρ	Water Density [$\frac{kg}{m^3}$]
ϕ	Sliding Mode Boundary Layer Thickness
ψ_{12}	Angle between Leader Surge Vector and l_{12} Vector [rad]
θ	Yaw Angle of Vessel [rad]
ω	Wake Factor

CHAPTER 1

INTRODUCTION AND LITERATURE REVIEW

Formation control is a type of autonomous control used to set and maintain a formation between two or more mobile platforms. There are many applications for this technology, both military and civilian, and most of the applications for this technology revolve around surveillance and remote sensing. One benefit of this technology is that it could enable a human operator to oversee and control an array of robotic sensors using formation control instead of a single unit as would be the traditional case, thus multiplying the effectiveness of the operator. This growing field encompasses a variety of application platforms ranging in scope from robotic automobiles [1] to satellites [2].

The platform this research focuses on is the marine surface vessel, which traditionally use only a propeller and rudder for control. Thus these surface vessels are usually underactuated (having fewer controlled degrees of freedom (DOF), than total DOF of the system) and non-holonomic (there are no dynamic constraints to assist in the control, effectively reducing the number of DOF's). Due to this, these vessels are difficult to control, methods such as feedback linearization can cause poor performance [3]. This differs from other surface vehicles such as cars, which are also underactuated, but are holonomic. This is due to the friction of the wheels preventing the car (under normal operation) from moving laterally. Regarding other methods of control for sur-

face marine vehicles, Roberts [4] covers the history and development of marine control methods from the development of PID control up to modern autopilot programs. Fossen [5] summarizes the current controller developments and covers details spanning marine vessel dynamics to wave disturbance simulation. Other methods of surface vessel control, not covered by Fossen, range from fuzzy logic [6], to linear quadratic Gaussian and model predictive control [7]. Naeem [7] also introduces a fault tolerant sensor fusion technique to the control of a physical boat. An alternative method is sliding mode control. Ashrafiuon [8] shows a trajectory tracking controller using this method. Fahimi [9] using this same method for control of the formation. It is this last method, sliding mode control, that is used for formation control in this research, however, formation control has its own procedural methods.

Prior work done on formation control is outlined by McDowell [10] and can be sorted into either of two types. The first is a hierarchal approach and the second is a leader-follower method. Of these methods the first, hierarchal has multiple sub methods whereas the leader follower method has basically two sub methods. These subtypes are for a single leader and follower, with variations on this, or for a multiple leader and/or with multiple followers.

The hierarchal, or behavior based approach, includes an additional component incorporated into the hierarchy of the trajectory controller. Of the hierarchal method there have been two approaches, firstly, a behaviour based approach that uses multiple behaviors, in conjunction and in competition, to set the desired trajectory. This method adds an additional behaviour, or weighted term, to the trajectory tracking controller, with one of these behaviors/weighted terms the position within the formation, verses obstacle avoidance, for example. This would incorporate the maintenance of the formation into the trajectory of the robot. This behavior architecture was used by Tomlin et al. [11] for clusters of unmanned aerial vehicles (UAV). An additional arial vehicle controller using this method is developed by Anderson [12] based on bi-

ology and using game functions to produce a cooperative behavior, in this case flock flight formations. Further, Balch et al [13] uses this method for formation control with land vehicles.

The second hierarchal approach is more serial. For example: first a trajectory is set for all units as a group. Then this trajectory is broken down into trajectories for each unit, and using a general trajectory controller for each unit, is used to produce the desired formation. Vaneck [6] does this with a way-point following controller using fuzzy logic, although not specifically designed for formation control.

The second method, using leaders and followers, can also be further broken down into two control schemes. The first would be based on a single leader and follower and uses a distance and angle to set the formation, the second scheme uses two distances between the follower and two leaders where one leader is actually a follower of the true leader. The leader follower method has been implemented with many variations such as applying graph theory as done by Desai [1]. Desai also used a decentralized control method, the alternative was accomplished by Kapila [2] earlier for the leader follower method. McDowell [10] used a genetic algorithm to develop a neural network controller based on biological examples to perform formation control using a leader follower approach. Although that method also fit into the hierarchal approach as this formation behavior was incorporated with other behaviors such as obstacle avoidance. The theoretical work done for surface marine vessels by Fahimi [9] forms the basis of this research. Additionally, the secondary scheme for leader follower method using two leaders and the respective distances for control has been investigated by Schoerling [14] in parallel with this research.

Of the above cited research there has been research into the control of marine vessels ([3], [5], [6], [7], [8], [9], [4], [15], [16], [17], [18]) and into various formation control methods ([1], [19], [20], [2], [13], [12], [10], [11]). Many of those focusing on control of marine craft have been focused on craft other than boats, namely sub-

mersibles, as they are not underactuated. Thus there has been a relative dearth of research covering both formation control and surface vessels. This research aims to fill this gap with experimentally results, as opposed to solely simulations as done in the majority of the cited studies. It should be noted that this controller is for a boat to follow another boat, and that through the virtualization of the leader boat, a trajectory controller based on sliding mode control is equivalently produced.

The rest of the research presented in this thesis covers the initial theoretical work, including a simple model and simulations of said model. The application model, with the complexities to match the physical vessel used, is presented as well. Simulations with this more complex model are also covered. The next chapter covers the design and implementation of the hardware required for the control of the boat. Furthermore, the software and programs (such as the controller, and software-hardware interface) are described. The proceeding chapter covers the implementation of these programs used for the identification of the boat parameters and closed loop testing, and include the obtained results. Concluding is a chapter covering additional work to improve the model and identification of the vessel to obtain better results.

CHAPTER 2

SIMPLIFIED SIMULATION DESIGN

The objective of this component of research, the preliminary design, is to gain an in depth understanding of the constitutive equations that make up the model of the boat. In addition, developing a controller based on this boat model, gives valuable experience and a framework on which later controllers are based.

This chapter covers the theory behind the model used to represent a generic boat and the controller used for this model. The boat model is used as the plant in the design of this control program to control the formation of two or more marine vehicles. While many control methods can be used for path following or trajectory tracking (refer to chapter 1), the sliding mode control scheme is best suited for formation control [3] and is focused on here. The reason sliding mode is better suited for formation control is that the algorithm produces a controller output per system output, and since surface marine vessels are underactuated (not having the same number of system outputs and controller inputs) this control scheme will not work for trajectory tracking as well as other control schemes would, as for example, a chain rule method [3]. The other benefits and reasons for choosing sliding mode control are that it has inherent robustness and is better able to handle model uncertainties as well as outside disturbances.

Further sections of this chapter recount the derivation of the boat model (plant),

simulations of this plant, and the derivation of the controller. Then the assembly of the complete system and simulations are described.

2.1 Dynamic Boat Plant Model

Previous work done by Fossen [5] relating to the development of the constitutive equations defining the behavior of a boat were used as a basis for the model. While the true boat model is a six degree of freedom (DoF) model, the simplified form is obtained by neglecting some nonlinear terms with the following assumptions. Specifically the heave, pitch, and roll DoF's are neglected. These DoF's are dropped to correspond to an ideal environment. That is, the water surface is assumed to be flat having no waves, thus leaving as the only DoF's remaining (u, v, r) which correlate to lengthwise, transverse, and rotational degrees of freedom respectively. The boat is also assumed to have longitudinal and lateral axes of symmetry, allowing the coupling terms in the mass and damping matrices to be disregarded. The equations defining this simplified model are as follows:

$$\begin{aligned}
 m_{11}\dot{u} - m_{22}vr + d_{11}u &= F \\
 m_{22}\dot{v} + m_{11}ur + d_{22}v &= 0 \\
 m_{33}\dot{r} + (m_{22} - m_{11})uv + d_{33}r &= T
 \end{aligned}
 \tag{2.1}$$

Here, the m_{ij} terms and d_{ij} terms are components of the mass and damping matrixes respectively. These two matrixes are diagonal as part of the simplifications mentioned previously (neglection of coupling terms). The u, v, r terms are the surge sway and yaw velocities respectively and the F and T are the force and torque applied to the vessel. For the control of the plant, the plant outputs needed to be in global terms for comparison with the lead boat, and to obtain the formation parameters. To

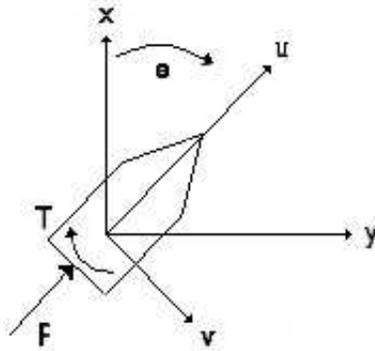


Figure 2.1: Boat Diagram

convert these equations from local to global coordinates, i.e. from (u, v, r) to $(\dot{x}, \dot{y}, \dot{\theta})$, transform relations are used. These transform relations are depicted graphically in Fig. 2.1, which shows the boat, driving force (F) and torque (T), and the local and global coordinates. Based on this figure, the following transformations can be written.

$$\begin{aligned}
 u &= \dot{x} \cos \theta + \dot{y} \sin \theta \\
 v &= -\dot{x} \sin \theta + \dot{y} \cos \theta \\
 r &= \dot{\theta}
 \end{aligned}
 \tag{2.2}$$

To get the global equations of motion Eq. 2.3 is differentiated and substituted into Eq. 2.2 and the resulting equations are solved for $(\ddot{x}, \ddot{y}, \ddot{\theta})$ and simplified by further substitutions of $m_r = \frac{m_{11}}{m_{22}}$ and $m_d = m_{22} - m_{11}$. Thus, the global equations of motion

are derived as follows:

$$\begin{aligned}\ddot{x} &= \frac{1}{m_{11}}(m_r d_{22} v \sin \theta - d_{11} u \cos \theta + \dot{\theta}(v \cos \theta - m_r u \sin \theta) m_d + F \cos \theta) \\ \ddot{y} &= \frac{1}{m_{11}}(-m_r d_{22} v \cos \theta - d_{11} u \sin \theta + \dot{\theta}(v \sin \theta + m_r u \cos \theta) m_d + F \sin \theta) \\ \ddot{\theta} &= \frac{1}{m_{33}}(-m_d u v - d_{33} \dot{\theta} + T)\end{aligned}\quad (2.3)$$

This can also be written in simplified state space form as:

$$\dot{\mathbf{q}} = f(\mathbf{q}, \mathbf{u}) \quad (2.4)$$

where $\mathbf{q} = [\dot{x}, x, \dot{y}, y, \dot{\theta}, \theta]^T$ and $\mathbf{u} = [F, T]^T$

Using the global equations of motion the plant model is created in Simulink with the following steps. First, the boat plant interface is created, consisting of two constant blocks representing the control inputs (Force and Torque), a subsystem block for the boat plant, and two output scopes to observe the plants reaction to the inputs. Refer to Fig. 2.2 showing the plant model.

The second step is to construct the plant subsystem. This subsystem block has two inputs, the force and torque, and six outputs, representing the state of the plant/boat. These outputs are the (x, y, θ) positions as well as their derivatives $(\dot{x}, \dot{y}, \dot{\theta})$. Inside the subsystem there are six more constant blocks corresponding to the diagonal elements of the mass and damping matrices. The values of these elements are collated to a single vector for diagram ashetics (Fig. 2.3). In addition to this, there are three embedded functions. The plant subsystems consists primarily of these three embedded MATLAB functions: “Theta Dynamics”, “X Dynamics”, and “Y Dynamics” which correlate with the equations of motion (Eq. 2.3). At the outputs of the three embedded functions are sets of two integrators to give the values for the velocity and position of each degree of freedom from the acceleration calculated by the embedded

functions. The initial conditions of the integrators are set as zero via constant blocks and can be changed to correspond to initial conditions of the system.

Refer to Fig. 2.3 for reference of the Simulink layout of the plant subsystem and refer to Eq. 2.3 for the contents of the embedded functions. Note that in Fig. 2.3, the values for the mass and damping elements are chosen based on values from Rayhanoglu [18]. Thus : $m_{11} = 200$ kg, $m_{22} = 250$ kg, $m_{33} = 80$ kg.m², $d_{11} = 70$ kg/s, $d_{22} = 100$ kg/s, and $d_{33} = 50$ kg.m²/s.

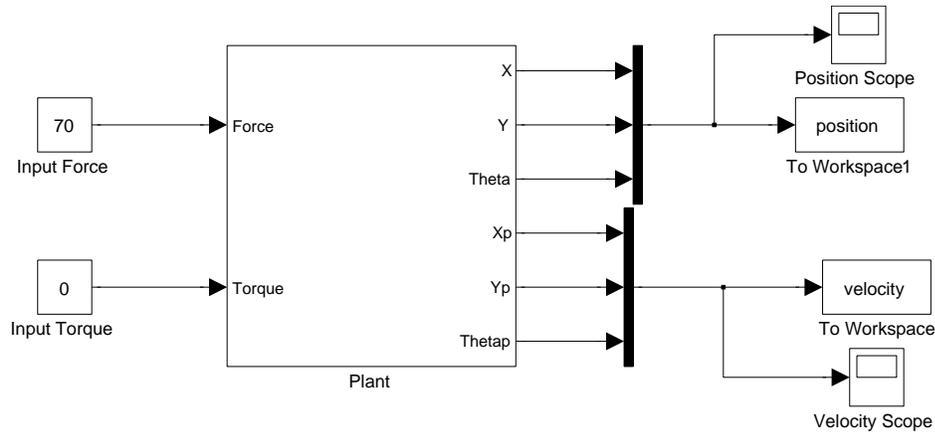


Figure 2.2: Boat Plant Interface

2.2 Simulation of Plant Model

To test the plant model three motion/input conditions are considered: straight - constant force and zero torque, circle(arc) - constant force and constant torque, and harmonic oscillations- constant force and alternating torque. These three motions are used as they make up the main components in common vessel maneuvers. That is, most common boat trajectories can be broken down into these three motions. For all cases the value chosen for the force was 70 N as this gives a final velocity of one meter per second due to the damping term d_{11} of 70 Ns/m. The following

Figures 2.4 through 2.8 show the response of the model to these three inputs. For the first simulation, the boat starts at the global origin with no initial velocity or acceleration and at an initial orientation of $\frac{\pi}{4}$ radians. The expected result was that the boat would accelerate up to a speed of 1 m/s going in a straight line along its initial orientation. As can be seen in Fig. 2.4 and Fig. 2.5 the simulation of the plant exhibits the expected responses.

For the second simulation, the circular case, the boats started at the origin with an initial orientation of 0 radians and zero velocity and acceleration. Here the force was 70 N, and the torque was 2.5 Nm. From this initial state the expected result of the simulation was that the boat would accelerate into a circular path. As observed in Fig. 2.6 and 2.7 this is the simulated result.

The third simulation has the boat starting at the origin with an initial orientation of $\frac{\pi}{5}$. The initial velocity is zero except the angular rate which started at -0.85219 radians per second. This starting state is to make the oscillations more symmetric about the x axis. Fig. 2.8 and 2.9 shows the simulated results and it can be seen that the expected behaviour of small oscillations along the x axis were obtained.

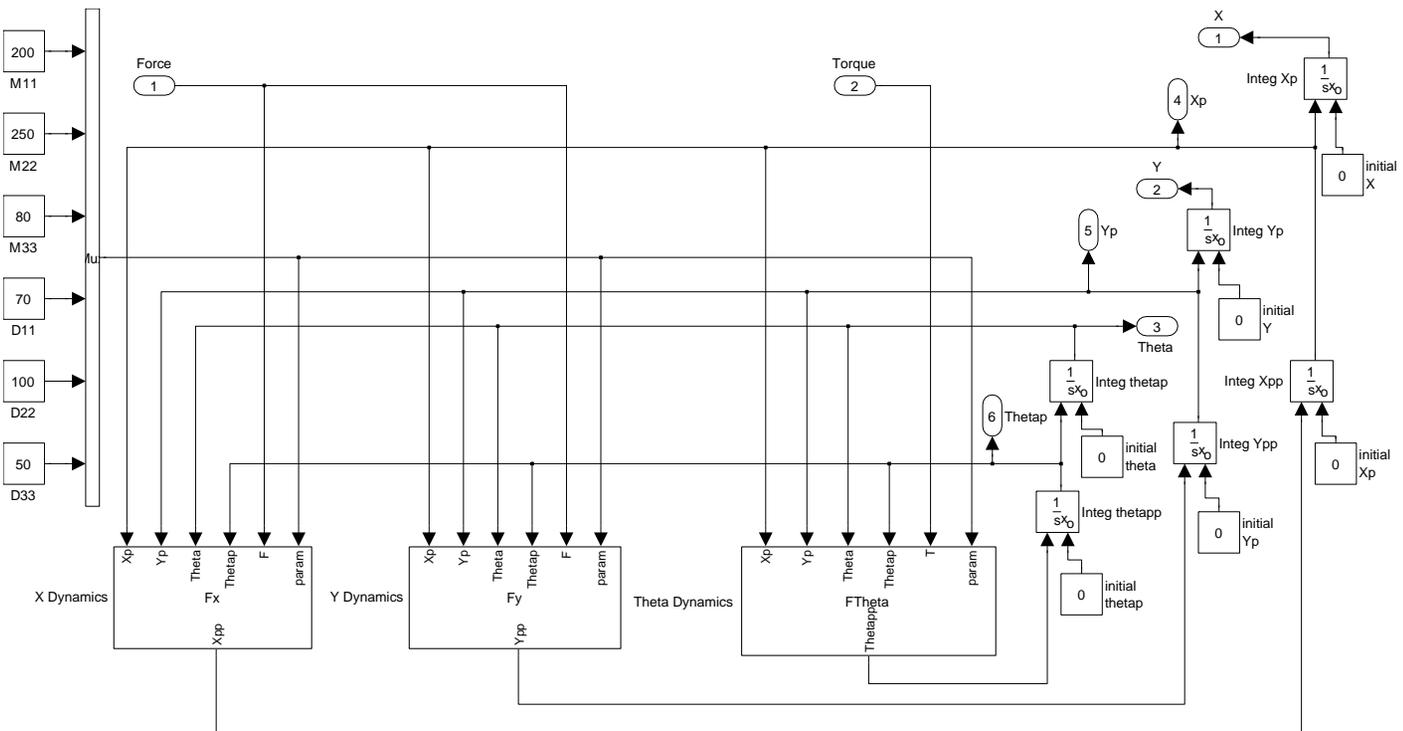
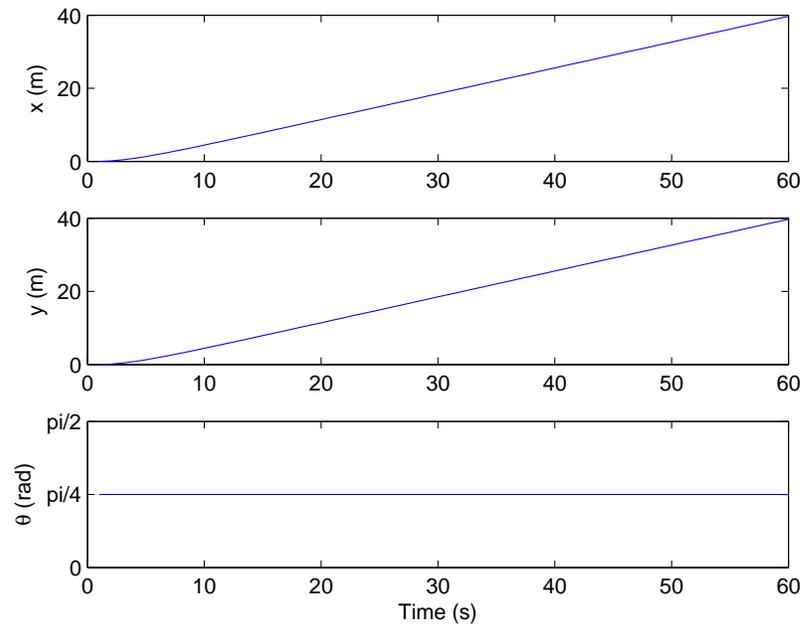
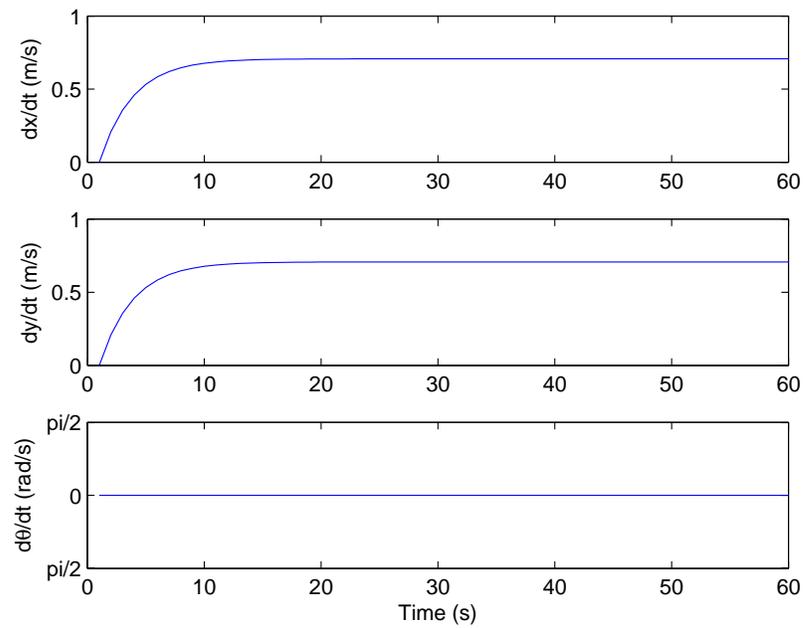


Figure 2.3: Boat Plant



(a)



(b)

Figure 2.4: Straight Line Boat Response (a) Position response; (b) Velocity response.

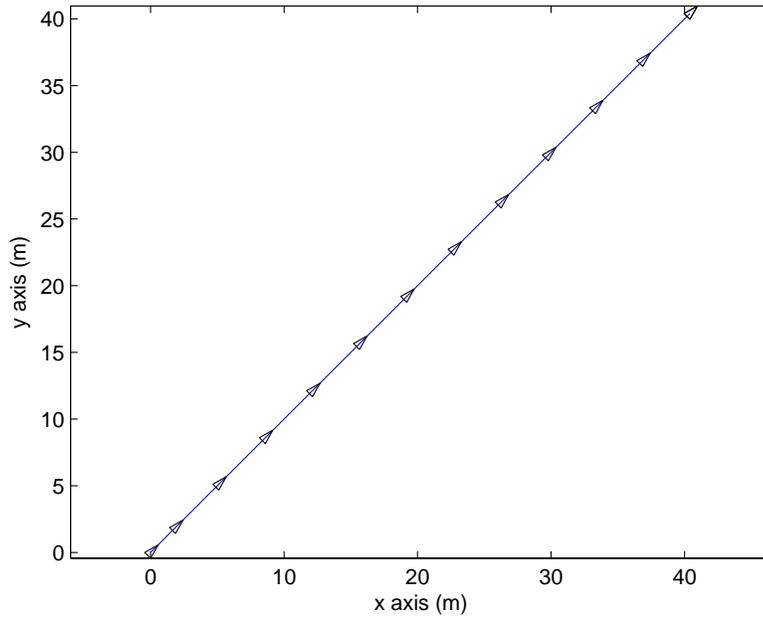
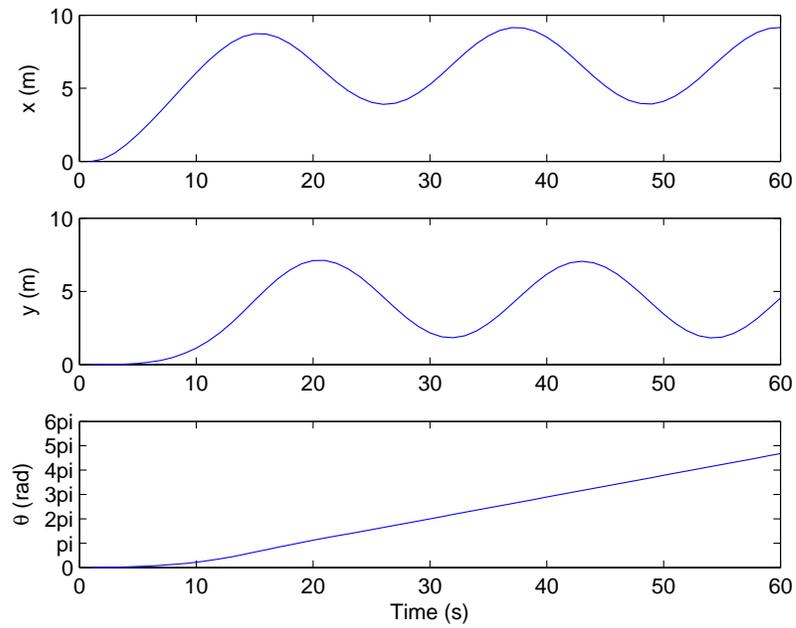
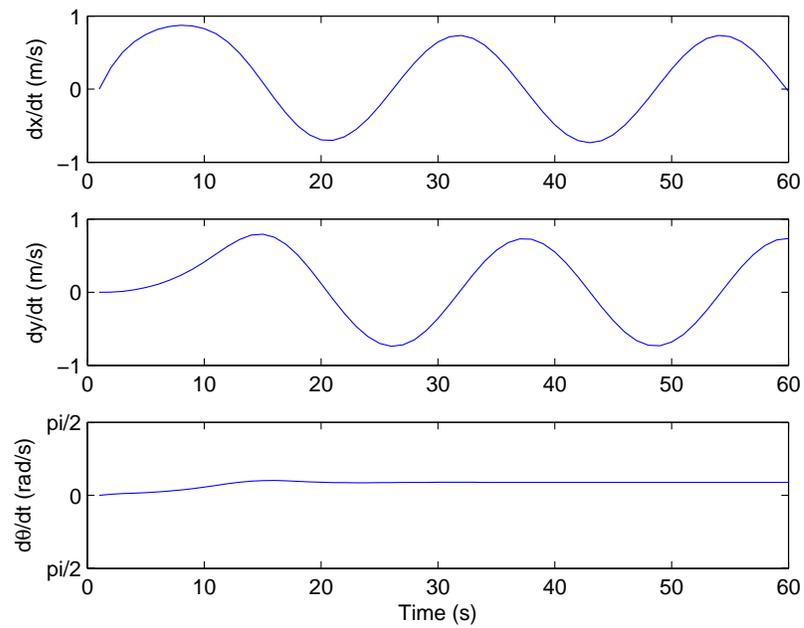


Figure 2.5: Straight Line Boat Response: Boat trajectory with orientation.



(a)



(b)

Figure 2.6: Circular Boat Response (a) Position response; (b) Velocity response.

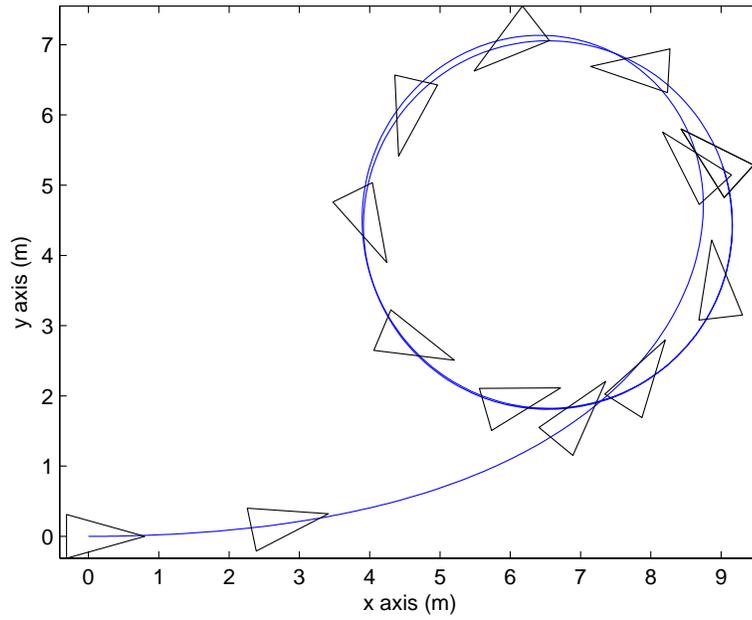
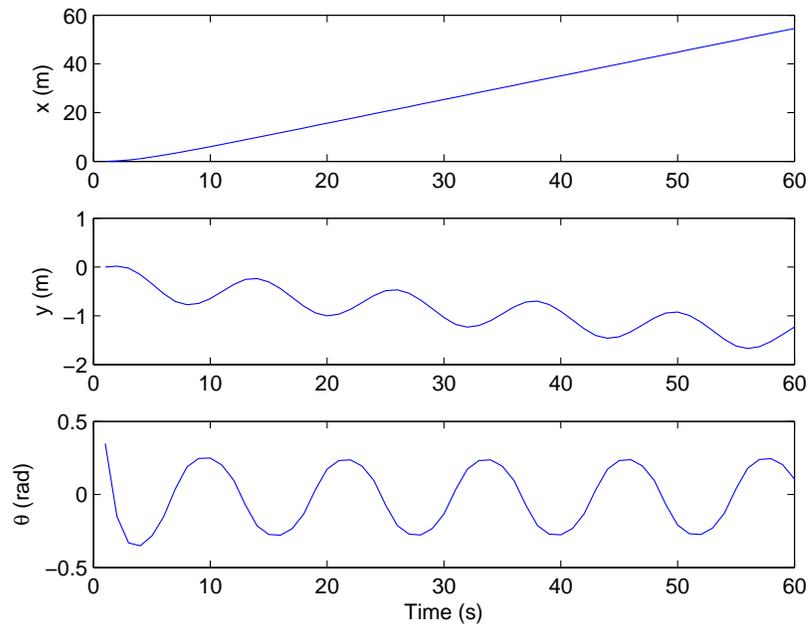
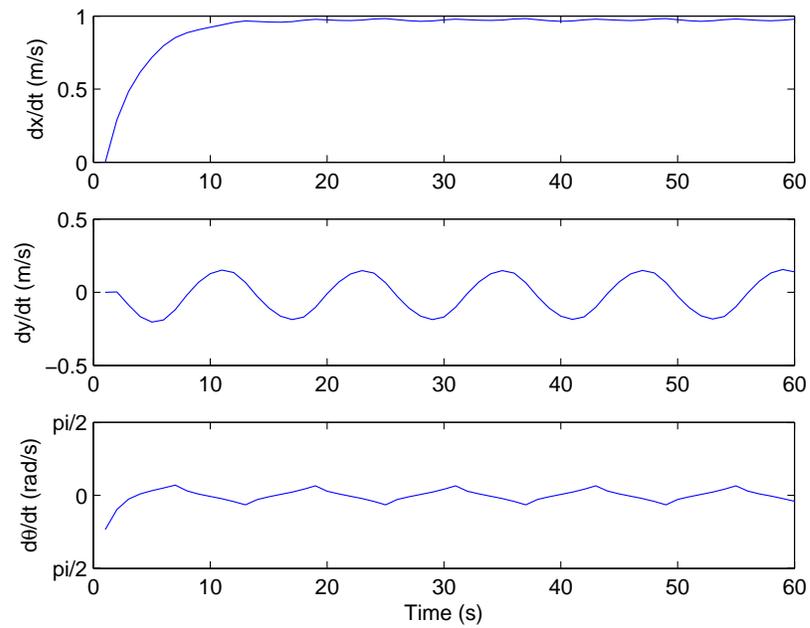


Figure 2.7: Circular Boat Response: Boat trajectory with orientation.



(a)



(b)

Figure 2.8: Sinusoidal Boat Response (a) Position response; (b) Velocity response.

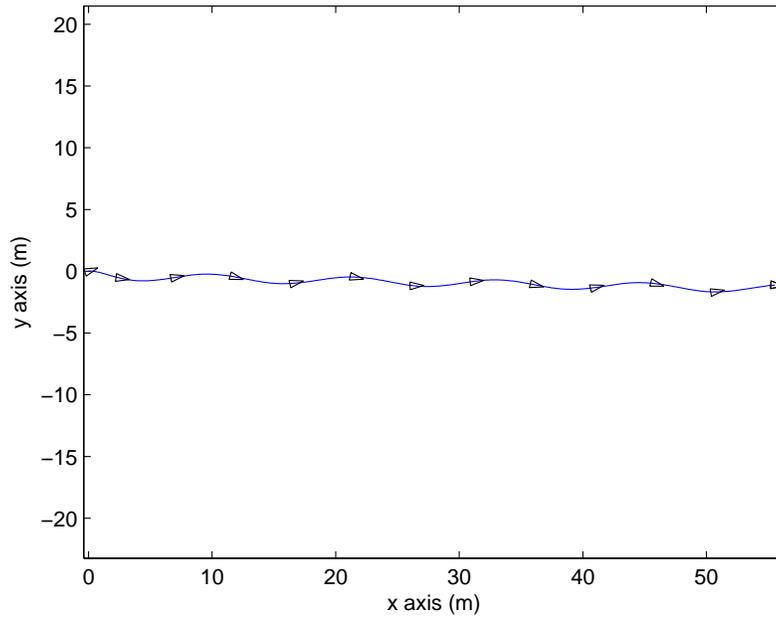


Figure 2.9: Sinusoidal Boat Response: Boat trajectory with orientation.

2.3 $l-\psi$ Controller Design

As discussed in chapter 1 formation control is the control of the internal geometry formed by the leader and follower vehicles. For simplification, only two boats were used for this controller design (a leader and a follower) thus the method used for the control of the boat is an $l-\psi$ control scheme. l is defined as the distance between the center of gravity (CG) of the lead boat and a control point ‘p’ on the follower boat. ψ is the relative angle between the boats and is depicted in Fig. 2.10. The reason a control point is used for the position of the follower boat and not the follower CG is because if the CG is used then the controller would only be able to control the position of the follower and would have no ability to limit the orientation. By using a control point that is some nonzero distance ahead of the CG the controller is able to set the orientation as well as the position of the follower. Thus the outputs of our leader-follower control system are l and ψ . Figure 2.10 also depicts the controlled formation parameters (outputs) as well as the position states of the leader and follower boat that are used to define the formation parameters. These outputs are calculated as follows:

$$l_{12} = \sqrt{(x_2 - x_1 + d \cos \theta_2)^2 + (y_2 - y_1 + d \sin \theta_2)^2} \quad (2.5)$$

$$\psi_{12} = \arctan\left(\frac{y_2 - y_1 + d \sin \theta_2}{x_2 - x_1 + d \cos \theta_2}\right) - \theta_1 \quad (2.6)$$

Here (x_1, y_1, θ_1) is the state of the lead boat and (x_2, y_2, θ_2) is the state of the follower boat. These parameters are shown in Fig. 2.10.

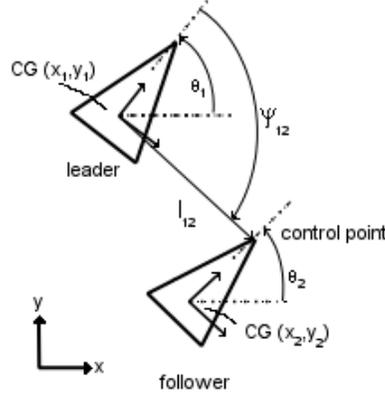


Figure 2.10: System States

The sliding mode control method [21] was used to derive the control law. In this method a time-varying surface is made such that $s(\mathbf{x}; t) = 0$ where:

$$s(\mathbf{x}; t) = \left(\frac{d}{dt} + \lambda\right)^{n-1} \tilde{\mathbf{x}} \quad (2.7)$$

and $\tilde{\mathbf{x}} = [l_{12} - l_{12}^d, \psi_{12} - \psi_{12}^d]^T$ is the tracking error of the system outputs, n is the order of the input-output model, and λ is a positive constant. For this case \mathbf{s} , the surface vector, is a vector made up of the two surfaces s_1 and s_2 . There are two surfaces for the boats formation control because there are two outputs (l, ψ). By definition, the dimension of \mathbf{s} is equal to the number of outputs for sliding mode control. Therefore, doing this method of control simplifies the tracking problem to remaining on the surface \mathbf{s} [21].

2.3.1 Input-Output Relations

However, to get this surface for a sliding mode controller the input-output relations of the system are required. The approach used is to find the $\ddot{l}, \ddot{\psi}$ relations from $(\ddot{x}, \ddot{y}, \ddot{\theta})$

using velocity and acceleration analysis [9]. This method is done, as opposed to taking the partial derivatives of the system outputs l, ψ and rearranging, for reasons of simplicity. This method, entails comparing the velocities at the control point p from the perspective of each vessels center of gravity (CG).

$$\text{follower point of view: } \bar{v}_p = \bar{v}_{cg_2} + \bar{\omega}_2 \times \bar{d} \quad (2.8)$$

$$\text{leader point of view: } \bar{v}_p = \bar{v}_{cg_1} + \dot{\bar{l}}_{12} + (\dot{\theta}_1 + \dot{\psi}_{12}) \times \bar{l}_{12} \quad (2.9)$$

Here the velocities of the CG's are $\bar{v}_{cg_1} = \dot{x}_1 \hat{i} + \dot{y}_1 \hat{j}$, and $\bar{v}_{cg_2} = \dot{x}_2 \hat{i} + \dot{y}_2 \hat{j}$. The subscript 1 indicates the leader boat, and the 2 indicates the follower. The other terms are defined as:

$$\begin{aligned} \bar{d} &= d \cos \theta_2 \hat{i} + d \sin \theta_2 \hat{j} \\ \bar{\omega}_2 &= \dot{\theta}_2 \hat{k} \\ \bar{l}_{12} &= l_{12} \cos(\theta_1 + \psi_{12}) \hat{i} + l_{12} \sin(\theta_1 + \psi_{12}) \hat{j} \\ \dot{\bar{l}}_{12} &= \dot{l}_{12} \cos(\theta_1 + \psi_{12}) \hat{i} + \dot{l}_{12} \sin(\theta_1 + \psi_{12}) \hat{j} \end{aligned}$$

Recall that d is the distance between the CG and the control point of the follower vessel. Rewriting these velocities in terms of global indexes yields:

$$\bar{v}_p = (\dot{x}_2 - \dot{\theta}_2 d \sin \theta_2) \hat{i} + (\dot{y}_2 + \dot{\theta}_2 d \cos \theta_2) \hat{j} \quad (2.10)$$

$$\begin{aligned} \bar{v}_p &= (\dot{x}_1 + \dot{l}_{12} \cos(\theta_1 + \psi_{12}) - (\dot{\theta}_1 + \dot{\psi}_{12}) l_{12} \sin(\theta_1 + \psi_{12})) \hat{i} \\ &+ (\dot{y}_1 + \dot{l}_{12} \sin(\theta_1 + \psi_{12}) + (\dot{\theta}_1 + \dot{\psi}_{12}) l_{12} \cos(\theta_1 + \psi_{12})) \hat{j} \end{aligned} \quad (2.11)$$

Combining the \bar{v} terms and solving for \dot{l} and $\dot{\psi}$, then simplifying the resultant equations yields the following:

$$\begin{aligned} \dot{l}_{12} = & ((\dot{y}_2 - \dot{y}_1) \sin(\theta_1 + \psi_{12}) + (\dot{x}_2 - \dot{x}_1) \cos(\theta_1 + \psi_{12}) \\ & + d\dot{\theta}_2 \sin(\theta_1 + \psi_{12} - \theta_2)) \end{aligned} \quad (2.12)$$

$$\begin{aligned} \dot{\psi}_{12} = & \frac{1}{l_{12}}(-(\dot{x}_2 - \dot{x}_1) \sin(\theta_1 + \psi_{12}) + (\dot{y}_2 - \dot{y}_1) \cos(\theta_1 + \psi_{12}) \\ & + d\dot{\theta}_2 \cos(\theta_1 + \psi_{12} - \theta_2) - l_{12}\dot{\theta}_1) \end{aligned} \quad (2.13)$$

Taking the time derivative and rearranging the above equations (2.12, 2.13) yields:

$$\begin{aligned} \ddot{l}_{12} = & ((\ddot{y}_2 - \ddot{y}_1) \sin(\theta_1 + \psi_{12}) + (\ddot{x}_2 - \ddot{x}_1) \cos(\theta_1 + \psi_{12}) \\ & + l_{12}(\dot{\theta}_1 + \dot{\psi}_{12})^2 + d\ddot{\theta}_2 \sin(\theta_1 + \psi_{12} - \theta_2) \\ & - d\dot{\theta}_2^2 \cos(\theta_1 + \psi_{12} - \theta_2)) \end{aligned} \quad (2.14)$$

$$\begin{aligned} \ddot{\psi}_{12} = & \frac{1}{l_{12}}(-(\ddot{x}_2 - \ddot{x}_1) \sin(\theta_1 + \psi_{12}) + (\ddot{y}_2 - \ddot{y}_1) \cos(\theta_1 + \psi_{12}) \\ & + d\ddot{\theta}_2 \cos(\theta_1 + \psi_{12} - \theta_2) + d\dot{\theta}_2^2 \sin(\theta_1 + \psi_{12} - \theta_2) \\ & - 2\dot{l}_{12}(\dot{\theta}_1 + \dot{\psi}_{12}) - l_{12}\ddot{\theta}_1) \end{aligned} \quad (2.15)$$

The second step is to take the relations between $(\ddot{x}, \ddot{y}, \ddot{\theta})$ and (F, T) and combine them with Eqs 2.14 and 2.15 above to get the $(\ddot{l}, \ddot{\psi})$ and (F, T) relations. \ddot{x} and \ddot{y} are substituted from the equations of motion (Eq. 2.3) into Eqs. (2.14, 2.15). Additionally, the angles are simplified by: $\alpha_0 = \theta_1 + \psi_{12}$ and $\gamma_1 = \theta_1 + \psi_{12} - \theta_2$, yielding:

$$\ddot{l}_{12} = \frac{1}{m_{11}}F \cos \gamma_1 + \frac{d}{m_{33}}T \sin \gamma_1 + f_l \quad (2.16)$$

$$\ddot{\psi}_{12} = \frac{1}{l_{12}} \left[-\frac{1}{m_{11}}(F \sin \gamma_1) + \frac{d}{m_{33}}T \cos \gamma_1 + f_\psi \right] \quad (2.17)$$

where:

$$f_l = \frac{f_x}{m_{11}} \cos \alpha_0 + \frac{df_\theta}{m_{33}} \sin \gamma_1 + \frac{f_y}{m_{11}} \sin \alpha_0 + g_l \quad (2.18)$$

$$f_\psi = -\frac{f_x}{m_{11}} \sin \alpha_0 + \frac{df_\theta}{m_{33}} \cos \gamma_1 + \frac{f_y}{m_{11}} \cos \alpha_0 + g_\psi \quad (2.19)$$

and:

$$g_l = -\ddot{y}_1 \sin \alpha_0 - \ddot{x}_1 \cos \alpha_0 + l_{12}(\dot{\alpha}_0)^2 - d\dot{\theta}_2^2 \cos \gamma_1 \quad (2.20)$$

$$g_\psi = \ddot{x}_1 \sin \alpha_0 - \ddot{y}_1 \cos \alpha_0 + d\dot{\theta}_2^2 \sin \gamma_1 - 2\dot{l}_{12}(\dot{\alpha}_0) - l_{12}\ddot{\theta}_1 \quad (2.21)$$

The input-output equations (2.16, 2.17) can also be written in the standard matrix form as:

$$\ddot{\mathbf{z}} = \mathbf{f} + \mathbf{b}\mathbf{u} \quad (2.22)$$

where:

$$\begin{aligned} \ddot{\mathbf{z}} &= \begin{bmatrix} \ddot{\psi}_{12} \\ \ddot{l}_{12} \end{bmatrix} & \mathbf{f} &= \begin{bmatrix} \frac{f_\psi}{l_{12}} \\ fl \end{bmatrix} \\ \mathbf{b} &= \begin{bmatrix} \frac{-\sin \gamma_1}{l_{12}m_{11}} & \frac{d \cos \gamma_1}{l_{12}m_{33}} \\ \frac{\cos \gamma_1}{m_{11}} & \frac{d \sin \gamma_1}{m_{33}} \end{bmatrix} & \mathbf{u} &= \begin{bmatrix} F \\ T \end{bmatrix} \end{aligned} \quad (2.23)$$

2.3.2 Sliding Mode Control Law

Using the sliding mode control method on Eq. 2.22, two stable surfaces (assuming $\lambda_{1,2} > 0$) are defined:

$$\mathbf{s} = \begin{bmatrix} s_1 \\ s_2 \end{bmatrix} = \begin{bmatrix} (\dot{\psi}_{12} - \dot{\psi}_{12}^d) + \lambda_1(\psi_{12} - \psi_{12}^d) \\ (\dot{l}_{12} - \dot{l}_{12}^d) + \lambda_2(l_{12} - l_{12}^d) \end{bmatrix} \quad (2.24)$$

or

$$\mathbf{s} = \dot{\mathbf{z}} - \mathbf{s}_r \quad (2.25)$$

where:

$$\mathbf{s}_r = \begin{bmatrix} \dot{\psi}_{12}^d - \lambda_1(\psi_{12} - \psi_{12}^d) \\ \dot{l}_{12}^d - \lambda_2(l_{12} - l_{12}^d) \end{bmatrix} \quad (2.26)$$

Using the derivative of Eq. 2.25 and substituting into Eq. 2.22, the required input \mathbf{u} for stable state outputs is:

$$\hat{\mathbf{u}} = \hat{\mathbf{b}}^{-1}(-\hat{\mathbf{f}} + \dot{\mathbf{s}}_r) \quad (2.27)$$

where the $(\hat{\cdot})$ indicates the nominal values of the matrices of Eq. 2.22.

This input is the nominal one, that is, the nominal control works only for a perfect model, solving with $\mathbf{s} = 0$ for $\hat{\mathbf{u}}$. To deal with non-zero \mathbf{s} and disturbance and uncertainty in the parameter values a gain term is added to the input as follows:

$$\mathbf{u} = \hat{\mathbf{b}}^{-1}(-\hat{\mathbf{f}} + \dot{\mathbf{s}}_r - \mathbf{k} \text{sign}(\mathbf{s})) \quad (2.28)$$

As the $\hat{\mathbf{b}}$ matrix is used as a denominator, it must be shown that the matrix is invertible, i.e. $\det(\hat{\mathbf{b}}) \neq 0$, at all times. The determinant can be simplified down to to:

$$\frac{1}{m_{11}m_{22}l_{12}} + \frac{dL}{m_{11}m_{33}l_{12}} \quad (2.29)$$

and as all the terms, d , L , l_{12} , and m_{ii} are greater than zero, the determinant is also nonzero as required.

The stability of the inputs is determined by the choice of controller gains and so the derivation of \mathbf{k} also proves stability. This is done by substitution of Eq. 2.28 into $\dot{\mathbf{s}}$, the derivative of Eq. 2.25, with the real parameters to get:

$$\dot{\mathbf{s}} = \mathbf{f} + \mathbf{b}\hat{\mathbf{b}}^{-1}(-\hat{\mathbf{f}} + \dot{\mathbf{s}}_r - \mathbf{k} \text{sign}(\mathbf{s})) - \dot{\mathbf{s}}_r \quad (2.30)$$

From this, the Lyapunov criteria, Eq. 2.31, [21] is applied to ensure stability.

$$\dot{\mathbf{s}} \cdot \mathbf{s} \leq -\eta \cdot |\mathbf{s}| \quad (2.31)$$

By rearranging Eq. 2.30 in to the following:

$$\dot{\mathbf{s}} = \left(\mathbf{f} - \mathbf{b}\hat{\mathbf{b}}^{-1}\hat{\mathbf{f}} \right) - \left(1 - \mathbf{b}\hat{\mathbf{b}}^{-1} \right) \dot{\mathbf{s}}_r - \mathbf{b}\hat{\mathbf{b}}^{-1}\mathbf{k} \text{sign}(\mathbf{s}) \quad (2.32)$$

multiplying by \mathbf{s} , and then inserting into the Lyapunov, with some simplifications such as $\text{sign}(\mathbf{s})\mathbf{s} = |\mathbf{s}|$, produces:

$$\mathbf{k} \geq \hat{\mathbf{b}}\mathbf{b}^{-1} \left[\left(\mathbf{f} - \mathbf{b}\hat{\mathbf{b}}^{-1}\hat{\mathbf{f}} \right) \frac{s}{|s|} - \left(1 - \mathbf{b}\hat{\mathbf{b}}^{-1} \right) \dot{\mathbf{s}}_r \frac{s}{|s|} + \eta \right] \quad (2.33)$$

Then to find \mathbf{k} , the maximum values of the right hand side of the equation were found, such as $\frac{s}{|s|} \leq 1$, and replacing $(\mathbf{f} - \hat{\mathbf{f}})$ with absolute values. Further, a positive and negative $\hat{\mathbf{f}}$ term are introduced for further simplification. Making these substitutions and rearranging to maximize the value, produced:

$$\mathbf{k} = \hat{\mathbf{b}}\mathbf{b}^{-1} \left(|\mathbf{f} - \hat{\mathbf{f}}| + \eta \right) + \left| 1 - \hat{\mathbf{b}}\mathbf{b}^{-1} \right| \left| \hat{\mathbf{f}} - \dot{\mathbf{s}}_r \right| \quad (2.34)$$

Using these controller gains insures stability. Using a saturation function instead of the sign function, Eq. 2.27 guarantees that the output trajectory reaches and remains on the surface ($s = 0$), that is, to reduce chattering. Since the surface (Eq. 2.24) is asymptotically stable for $s = 0$, the outputs approach and remain on the desired

point (l_{12}^d, ψ_{12}^d) . However, due to the reaching phase and sign function, the outputs will repeatedly cross the surface. This chattering can be removed by replacing the sign function with a saturation one. Thus, the last term of Eq. 2.28 is changed to:

$$\mathbf{k} \text{ sat}\left(\frac{\mathbf{s}}{\boldsymbol{\phi}}\right) = \begin{bmatrix} k_1 \text{sat}(s_1/\phi_1) \\ k_2 \text{sat}(s_2/\phi_2) \end{bmatrix} \quad (2.35)$$

Here $k_{1,2}$ are the controllers gains, and $\phi_{1,2}$ are the boundary layers of the surface $\mathbf{s} = 0$.

2.3.3 Simulink Implementation

The implementation of the controller is done in Simulink as with the plant. Refer to Fig. 2.11, the controller schematic. The controller Simulink block consists of five main embedded functions. Two of them, “a output” and “g output” take the system states as inputs and returns the angles γ_1 and α_0 . The “s output” function takes the outputs of the system and returns the values of s_1, s_2 (Eq. 2.24) as well as \mathbf{s}_r (Eq. 2.26) that correspond to the last and second to last terms of Eq. 2.28 [9]. The “b f outputs” function return the matrices correlating to the terms \mathbf{b}, \mathbf{f} from Eq. 2.23 according to the definitions in Eq. 2.18 and 2.19. The last sub-function takes all these matrices and combined them, as in Eq. 2.28 calculating the required inputs $u (F, T)$ for the boat plant.

Since the system outputs are used as variables to all the sub-functions of the controller, the Simulink block that calculates them is described next. This block takes the states of the leader and follower boats and returns the control systems outputs l_{12}, ψ_{12} , and their first order derivatives \dot{l}_{12} , and $\dot{\psi}_{12}$. This is done with three embedded functions. The first one, “Length and Angle” returned the two non

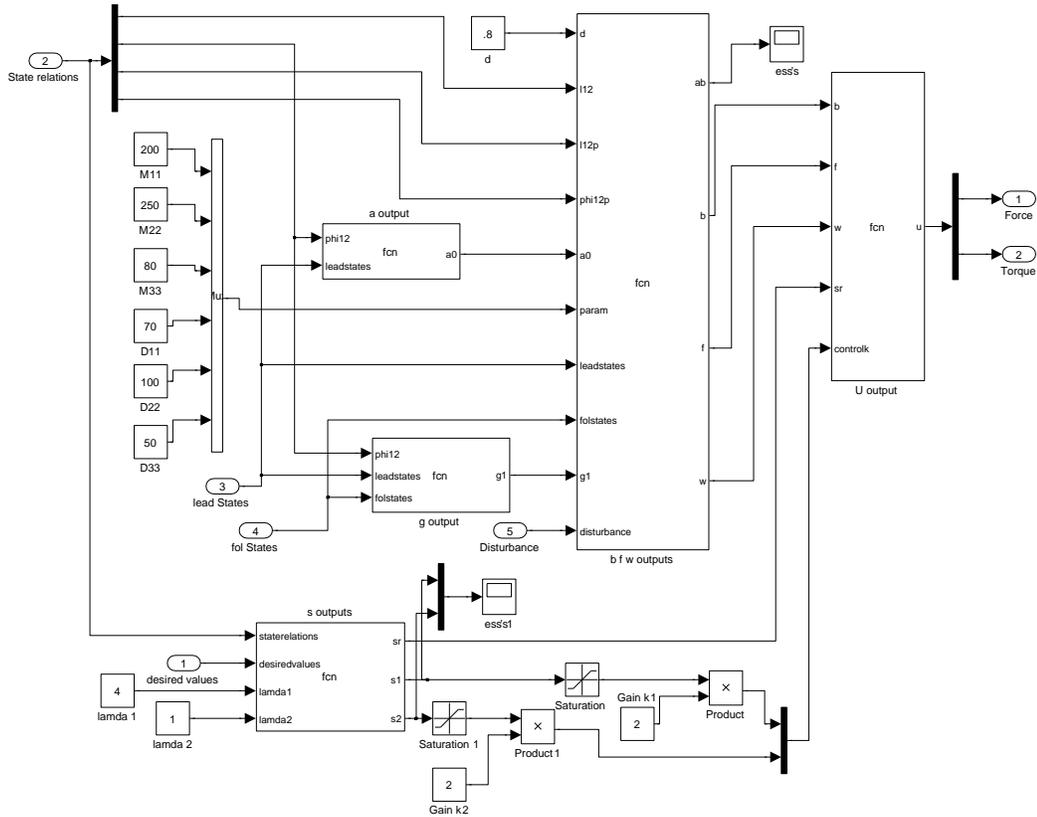


Figure 2.11: Controller Diagram

derivative parameters using Eq. 2.5 and 2.6. The other two returns the first derivative of the formation parameters l_{12} and ψ_{12} , implementing Eqs. 2.12 and 2.13. Refer to Fig. 2.12 for the block diagram of the output calculator component of the system controller.

2.4 Overall System

2.4.1 Lead Boats

Upon completion of the plant, output calculator, and controller, the last needed component to complete the system is the lead boat path generator. The reason for the lead boat trajectory generator is that a virtual lead boat is required for the formation

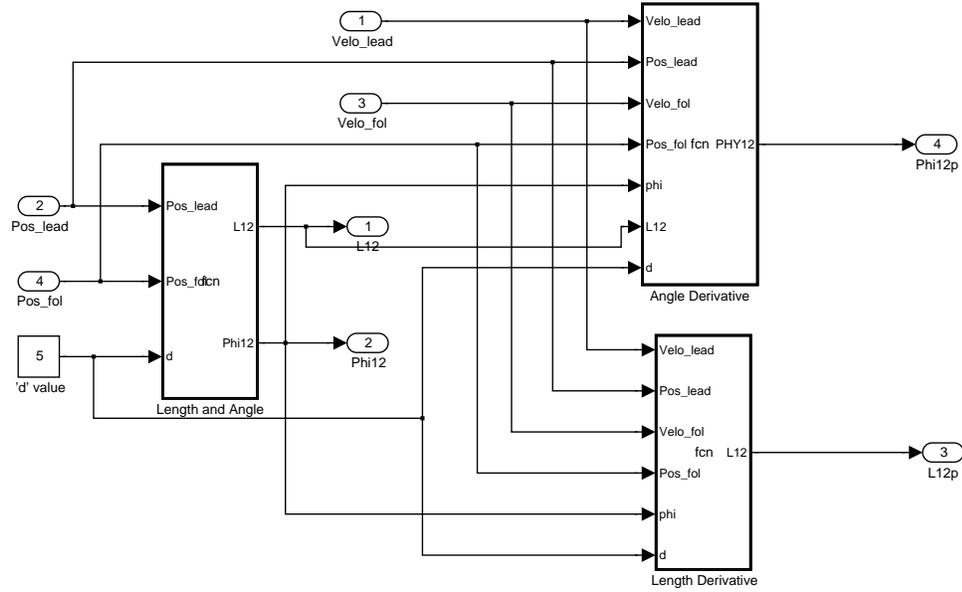


Figure 2.12: Formation Parameters (controller outputs) Calculations

controller, as it is impossible to make a formation with just a single boat. Three lead boat blocks are used in the Simulink system and each lead block corresponds to a certain trajectory. While any trajectory can be used for the leader boat, the three chosen provided the option of a straight path, a circular path, or a sinusoidal path. These three trajectories are chosen, for the same reason as they were the three test inputs for the initial plant simulations, as they can be combined to produce most possible trajectories (at least common ones). All three of these blocks have the same outputs: lead position, lead velocity, and “leadstates” (a data vector combining positions, velocities, and accelerations).

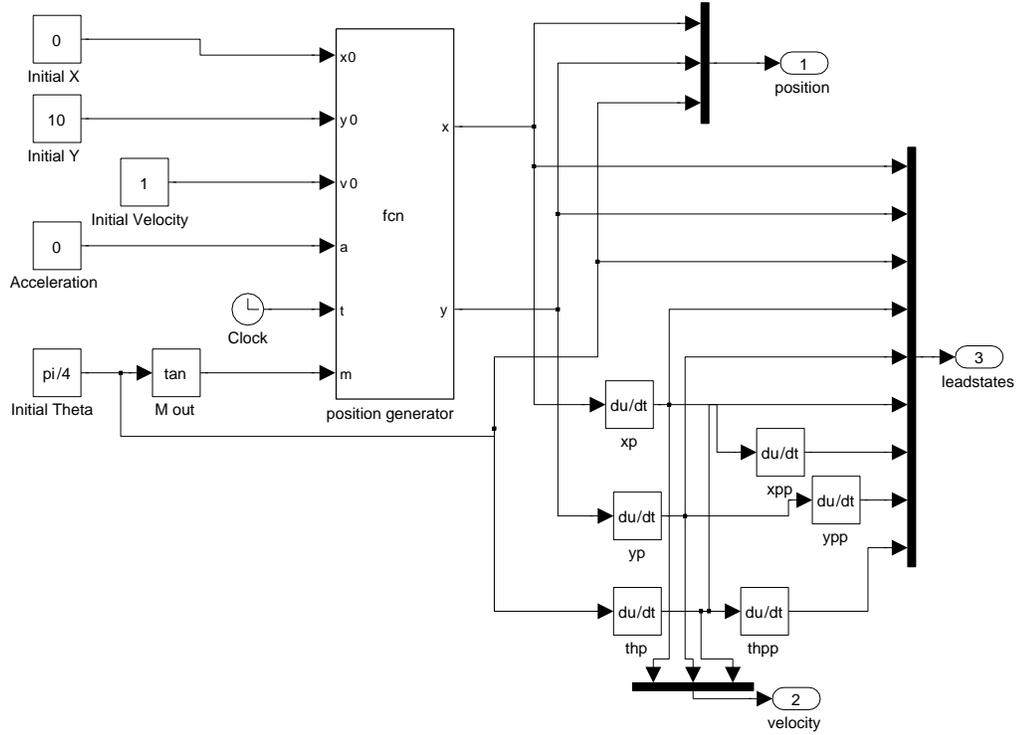


Figure 2.13: Lead Boat: Straight trajectory

2.4.2 Straight Line Trajectory

For the straight trajectory lead block, the equations used to define the motion path are:

$$x = \frac{1}{2}at^2 + v_i t + x_i \quad (2.36)$$

$$y = m(x - x_i) + y_i \quad (2.37)$$

$$\theta = \theta_i = \text{constant} \quad (2.38)$$

where the x_i , y_i , v_i , and θ_i are respectively the initial position, velocity, and orientation of the lead boat. a is the acceleration, and t is time. Refer to Fig. 2.13 for the corresponding Simulink block diagram.

2.4.3 Circular Trajectory

The second lead boat trajectory is the circular one. For this path of motion, the equations used are:

$$\theta_r = \frac{1}{2}\alpha_r t^2 + w_0 t + \theta_0 \quad (2.39)$$

$$x = x_i + r \cos(\theta_r) \quad (2.40)$$

$$y = y_i + r \sin(\theta_r) \quad (2.41)$$

$$\theta = \theta_r + \text{sign}(v_i) \frac{\pi}{2} \quad (2.42)$$

Here θ_r indicates the radial angle and α_r , w_0 are the angular acceleration and initial rotational velocity, and θ_0 is the initial radial angle. As for the previous trajectory x_i and y_i are the initial position. The block diagram of this lead boat motion generator is shown in Fig. 2.14.

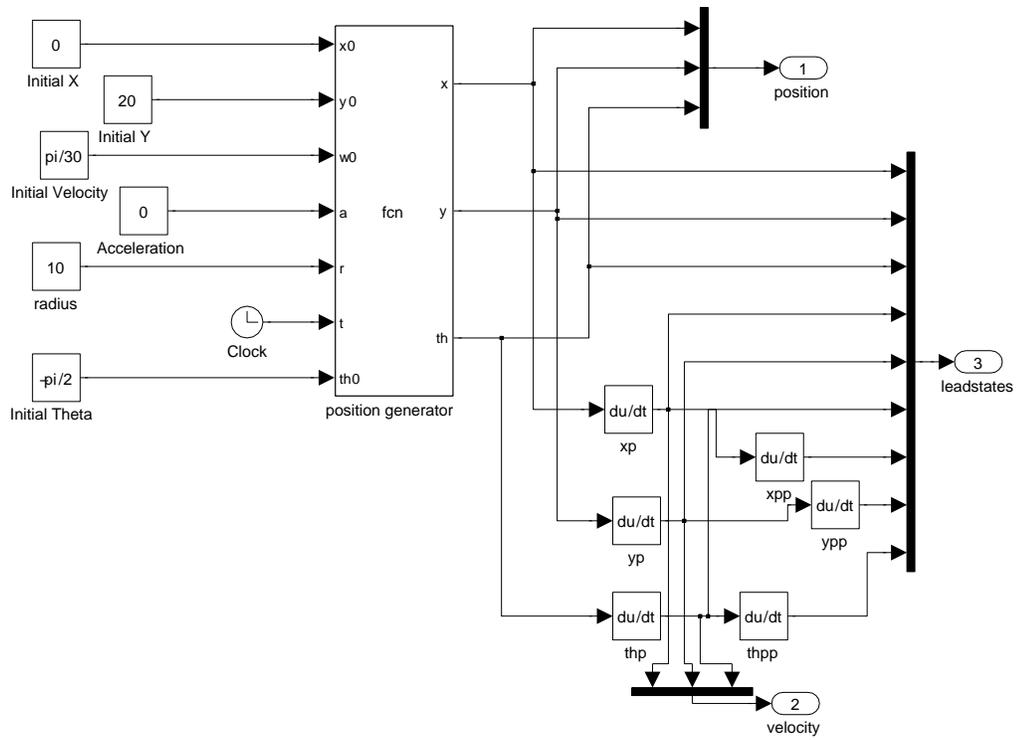


Figure 2.14: Lead Boat: Circular trajectory

2.4.4 Sinusoidal Trajectory

Lastly, the sinusoidal motion of the lead boat. For this motion, the equations that defined the path are:

$$L = \frac{1}{2}at^2 + v_it \quad (2.43)$$

$$H = 1.5 \sin(wt) \quad (2.44)$$

$$x = x_i + L \cos(\theta_i) + H \cos(\theta_i + \frac{\pi}{2}) \quad (2.45)$$

$$y = y_i + L \sin(\theta_i) + H \sin(\theta_i + \frac{\pi}{2}) \quad (2.46)$$

$$\theta = \theta_i + \frac{\pi}{4} \cos(wt + \frac{\pi}{16}) \quad (2.47)$$

Here the L and H terms make an arbitrary sinusoidal path. These are then rotated by θ_i to the desired orientation. The v_i term is the initial x axis velocity prior to rotation, a the corresponding acceleration, and w the angular rotation of the sinusoid oscillation. The diagram of this Simulink block is shown in Fig. 2.15.

2.4.5 Complete System Assembly

When these lead boats are combined with the other elements, the controller, the plant, and the formation parameter computing algorithm, it completes the system. The complete system diagram can be seen in Fig. 2.16. The system as a whole consisted of the controller whose outputs of “Force” and “Torque” are the inputs to the boat plant. From the plant the position and velocity are produced. This is combined with the position and velocity outputs of the lead boat and fed into the formation parameter calculator whose output are the system formation parameters l_{12} and ψ_{12} and their derivatives. These parameters and the desired parameter values are entered into the controller to complete the feedback loop. The leader and follower

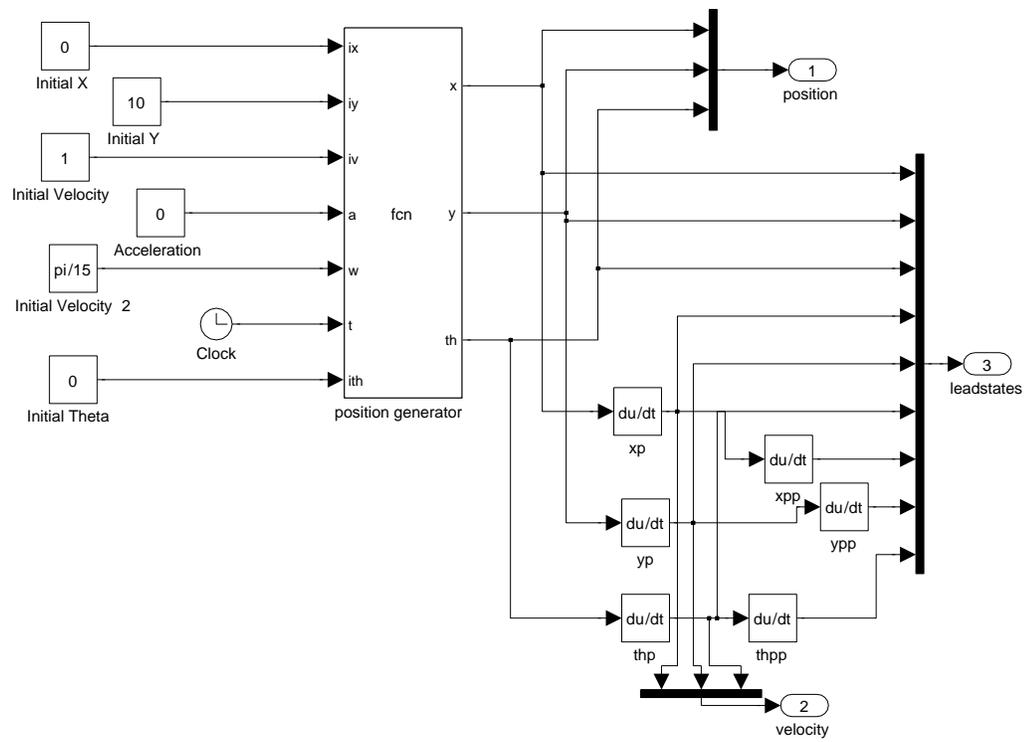


Figure 2.15: Lead Boat: Sinusoidal trajectory

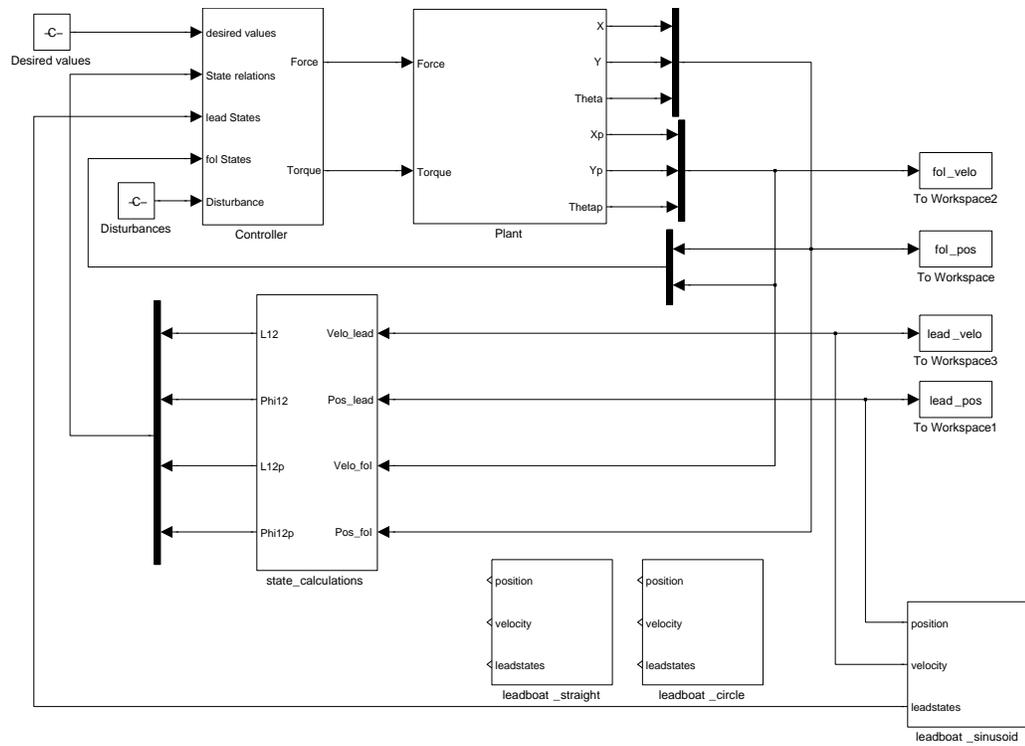


Figure 2.16: Overall System Diagram

positions and velocities are also copied to the MatLab workspace for recording and further processing.

2.5 Simulation and Discussion

2.5.1 Simulation of the closed-loop system

For the simulation of the system as a whole, and testing of the controller, the model is run for each version of the leader boat. Thus three simulations follow. For all the simulations the desired formation parameters are: $l_{12}^d = 10$ meters and $\psi_{12}^d = \frac{3\pi}{4}$ radians, thus the follower should have reach steady-state in a position to the right and slightly behind the leader.

As mentioned previously, the positions and velocities of the leader and follower boats are sent to the MatLab workspace. This allows post processing in the form of an m-file that plots both of the boats on the same plot as little boat icons, indicating both the positions and orientations of the boats at a set interval (every five seconds), along the lines indicating the respective leader and follower trajectories. The following Figs. 2.17, 2.18, and 2.19 show the results of these simulations.

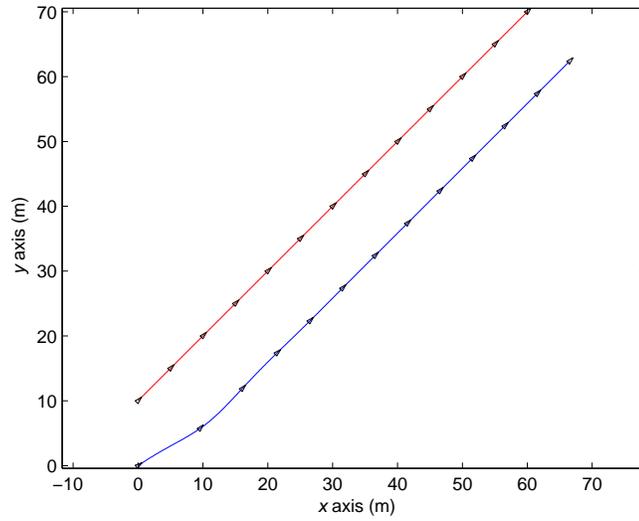


Figure 2.17: System Response: Straight lead path Follower begins at origin and lead boat starts in (x, y) position of $(0, 10)$

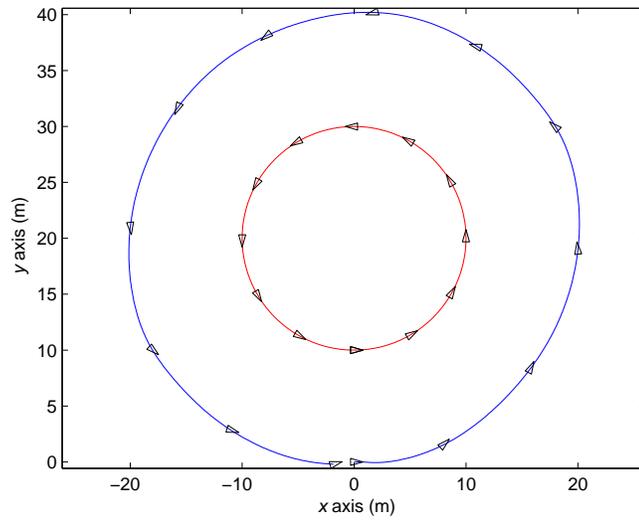


Figure 2.18: System Response: Circular lead path Lead boat starts at $(0, 10)$, follower starts at origin

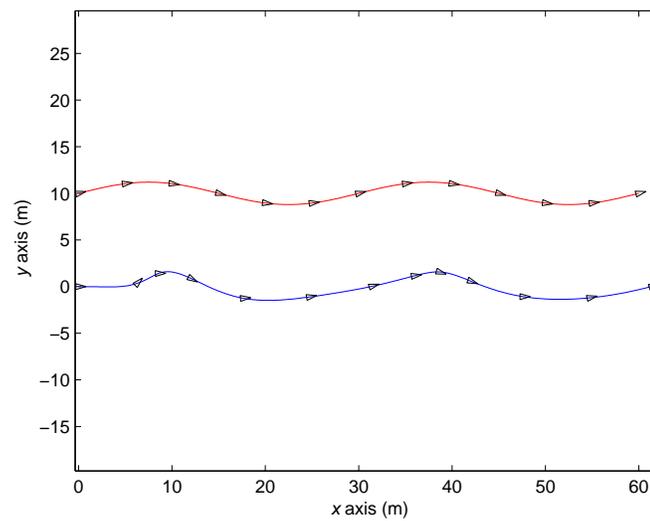


Figure 2.19: System Response: Sinusoidal lead path Lead boat begins at position (0,10), follower starts at origin

2.5.2 Discussion

From these figures it can be observed that the controller is able (within the model constraints) to move, then keep the follower boat near to the desired position with respect to the leader boat. Note that for the straight case and circular case the initial conditions are such that there are errors in the surfaces \mathbf{s} . In the straight case the controller overcomes this and moves the follower such that it reaches the desired position and orientation with respect to the leader and then follows the lead boat trajectory with that formation. In the circular case the controller does not maintain the desired position as the lead boat was changing position and orientation in such a fashion that the controller and follower boat are not able to reduce the error in position and velocity of the formation outputs due to saturation. This is rectified in later chapters through changes made to the lead boat paths and tuning of the control gains.

Of further note is that the orientation of the follower boat is not the same as the tangent to the path. This indicates that although the controller is not directly controlling the orientation of the follower boat, the orientation is inherently stable. This can be readily seen in the harmonic case Fig. 2.19. Since the control scheme used does not actually control the orientation of the follower, instead it is actually controlling the position of the control point with respect to the leader. Due to the use of the control point, the stability and equilibrium of this degree of freedom (orientation) depend upon the vehicles inherent stability, a consequence of the hydrodynamics and boat hull design [5].

In systems controlled using the sliding mode method the system outputs exhibit two distinct phases, a reaching phase and a sliding phase [21]. For this system specifically, the reaching phase can be observed in the system outputs l and ψ in Fig. 2.20. For the other figures of the system outputs Fig 2.21 and 2.22, this is not as appar-

ent, especially the sinusoidal case where the leader motion causes the control point to move near or behind the CG of the follower. However, the controller still moves towards the desired values and the oscillations in l and ψ can be observed, in Fig. 2.22, to settle towards the desired over the short time of the simulation.

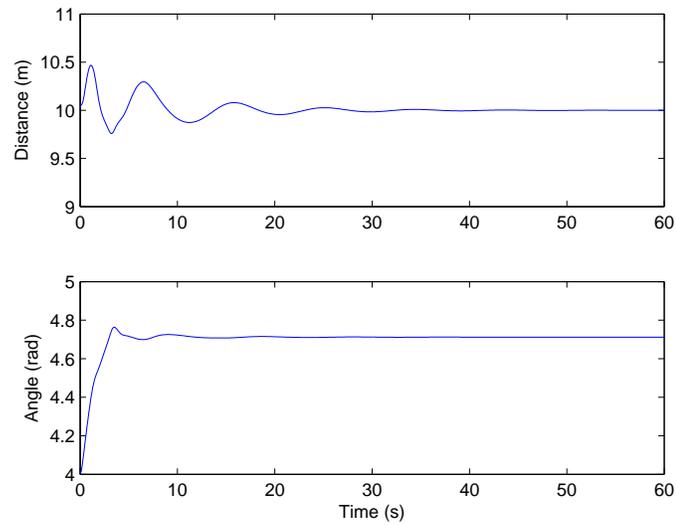
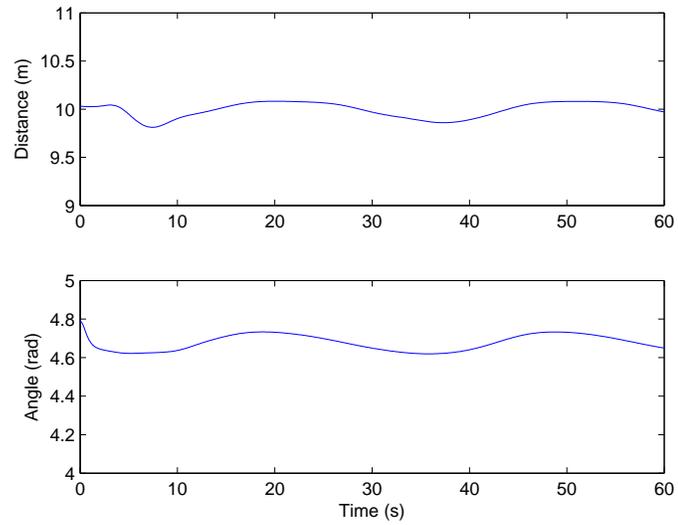
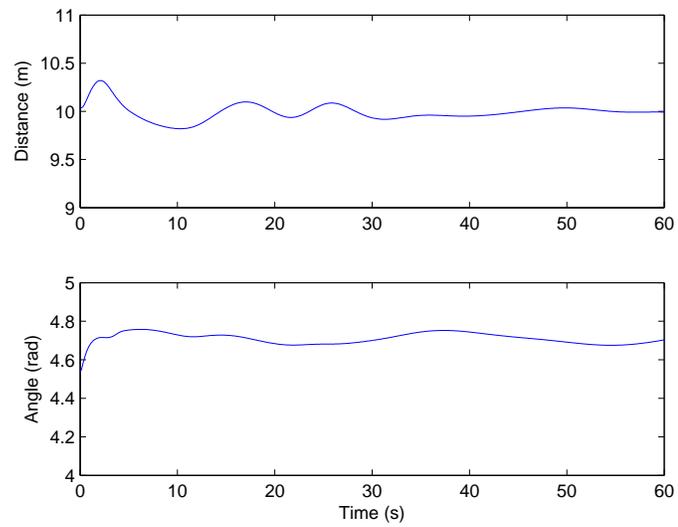


Figure 2.20: System Response: Straight line case system outputs l_{12} and ψ_{12}

These simulation results are covered in more detail in the chapter 3, as similar lead boat paths and simulation results are used and obtained respectively.

Figure 2.21: System Response: Circular case system outputs l_{12} and ψ_{12} Figure 2.22: System Response: Sinusoidal case system outputs l_{12} and ψ_{12}

CHAPTER 3

APPLICABLE DESIGN

Based on the preliminary design from the previous chapter, a more complex boat model is developed. The goals of this new model are to match as closely as possible the physical behavior of the vessel to be used for later real world testing. Through tweaking the plant model and then going through the same process of controller development, a controller is produced that is able to transfer from model to actual system without any radical adjustment. The process and models used are covered further in this chapter. Specifically the derivation of the equations of motion for the boat model in terms of global values, based on the loading parameters of the actual boat. Also covered is a review of the leader follower relations used for the controller, and the control laws used are also derived. The control laws and boat dynamics are simulated and the results are displayed and discussed.

3.1 Dynamic Model: Plant Equations Derivation

Based on the previous work by Fossen [5] and the preliminary design done in Chapter 2, a new dynamic model of the boat plant is developed. This new plant is made to correlate to the actual boat, as opposed to a generic case as was done for the preliminary design. To do this, the loading conditions in the equations of motion are changed to match the loading conditions of the real boat. Thus the general force and

torque are replaced with specific surge and sway forces and a yaw moment. Thus the dynamic equations of motion are:

$$\begin{aligned}
 m_{11}\dot{u} - m_{22}vr + d_{11}u &= X \\
 m_{22}\dot{v} + m_{11}ur + d_{22}v &= Y \\
 m_{33}\dot{r} + (m_{22} - m_{11})uv + d_{33}r &= N
 \end{aligned} \tag{3.1}$$

Here the loading forces are:

$$\begin{aligned}
 \text{Surge: } X &= T_h \cos(\alpha) \\
 \text{Sway: } Y &= T_h \sin(\alpha) \\
 \text{Torque: } N &= -T_h L \sin(\alpha) = -LY
 \end{aligned} \tag{3.2}$$

where L is the distance between the propellers and the CG. These loading conditions are obtained from the design of the actual boat. The layout of the propellers and steering system are shown in Fig. 3.1 as well as the corresponding simplified loading diagram. In these loading conditions the thrust term is defined as: $T_h = \rho D^4 K_T |n| n$, where n is the propeller speed in rotations per second. As in the initial design, the local velocities are related (Eq. 2.3) to global values and the derivatives $\dot{u}, \dot{v}, \dot{r}$ are then substituted into the equations of motion yielding:

$$\begin{aligned}
 m_{11}(\ddot{x} \cos \theta + \ddot{y} \sin \theta + v\dot{\theta}) - m_{22}v\dot{\theta} + d_{11}u &= X \\
 m_{22}(-\ddot{x} \sin \theta + \ddot{y} \cos \theta - u\dot{\theta}) + m_{11}u\dot{\theta} + d_{22}v &= Y \\
 m_{33}(\ddot{\theta}) + (m_{22} - m_{11})uv + d_{33}\dot{\theta} &= N
 \end{aligned} \tag{3.3}$$

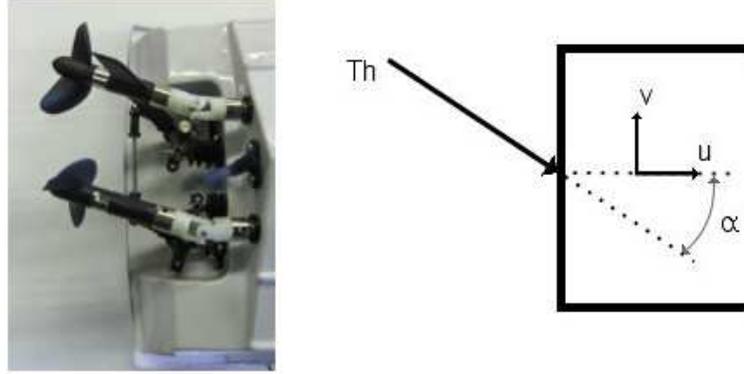


Figure 3.1: Boat Loading Diagram

This is then rearranged to solve for the terms \ddot{x} , \ddot{y} , $\ddot{\theta}$ as follows:

$$\begin{aligned}\ddot{x} &= \frac{\cos \theta}{m_{11}} \left[(m_{22} - m_{11})v\dot{\theta} - d_{11}u + X \right] - \frac{\sin \theta}{m_{22}} \left[(m_{22} - m_{11})u\dot{\theta} - d_{22}v + Y \right] \\ \ddot{y} &= \frac{\sin \theta}{m_{11}} \left[(m_{22} - m_{11})v\dot{\theta} - d_{11}u + X \right] + \frac{\cos \theta}{m_{22}} \left[(m_{22} - m_{11})u\dot{\theta} - d_{22}v + Y \right] \\ \ddot{\theta} &= \frac{1}{m_{33}} \left[-(m_{22} - m_{11})uv - d_{33}\dot{\theta} + N \right]\end{aligned}\quad (3.4)$$

Again, Simulink is used to make the boat plant, however for this model the interface is simplified to a single embedded function that produced the acceleration values. Refer to Fig. 3.2 showing the plant model diagram. The source code of the embedded function can be found in Appendix A. Note that the values for the mass and damping matrixes have new values. These values are chosen to better match the actual boat (a small RC model) values, where as the prior values were for simulating a freighter [18].

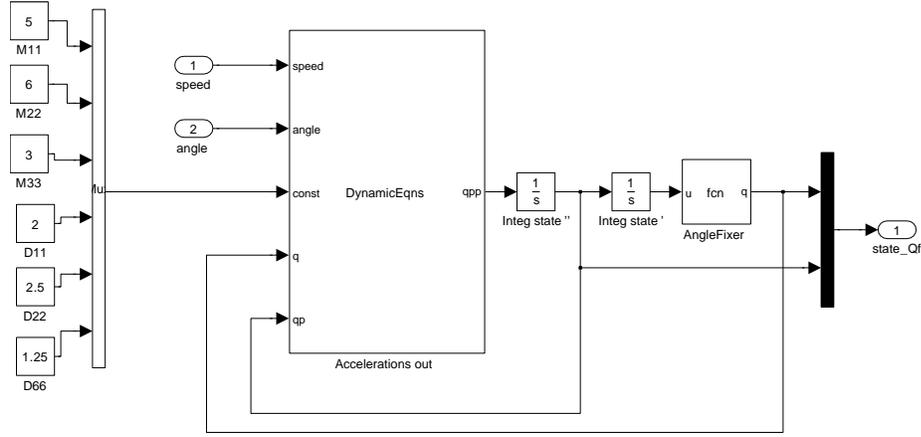


Figure 3.2: New Boat Plant Diagram

3.2 L-Psi Controller Design: Controller Derivation

The derivation of the control scheme for the leader and follower boats is done similarly to the derivation in chapter 2. First the formation parameters, that is the distance and angle between the center of gravity (CG) of the leader and the control point of the follower, are calculated. These two relations are:

$$l_{12} = \sqrt{(x_2 - x_1 + d \cos \theta_2)^2 + (y_2 - y_1 + d \sin \theta_2)^2} \quad (3.5)$$

$$\psi_{12} = \text{atan2} \left(\frac{-(x_2 + d \cos \theta_2) \sin \theta_1 + (y_2 + d \sin \theta_2) \cos \theta_1 - y_1 \cos \theta_1 + x_1 \sin \theta_1}{(x_2 + d \cos \theta_2) \cos \theta_1 + (y_2 + d \sin \theta_2) \sin \theta_1 - x_1 \cos \theta_1 - y_1 \sin \theta_1} \right)$$

Note that for this applicable design, the angle is defined in local terms only, where as the previous definition for ψ (Eq. 2.6) used global values. This is to prevent the controller from having to deal with the follower looping around the leader and getting angular values greater than 2π . Again the derivative of the relations are:

$$\dot{l}_{12} = (\dot{x}_2 - \dot{x}_1) \cos(\theta_1 + \psi_{12}) + (\dot{y}_2 - \dot{y}_1) \sin(\theta_1 + \psi_{12}) + d\dot{\theta}_2 \sin(\theta_1 + \psi_{12} - \theta_2) \quad (3.6)$$

$$\dot{\psi}_{12} = \frac{1}{l_{12}} \left(-(\dot{x}_2 - \dot{x}_1) \sin(\theta_1 + \psi_{12}) + (\dot{y}_2 - \dot{y}_1) \cos(\theta_1 + \psi_{12}) + d\dot{\theta}_2 \cos(\theta_1 + \psi_{12} - \theta_2) - l_{12}\dot{\theta}_1 \right)$$

and the second derivatives, with simplifications, are calculated using velocity and acceleration analysis, as in the initial design.

$$\begin{aligned} \ddot{l}_{12} &= \ddot{y}_2 \sin \phi + \ddot{x}_2 \cos \phi + d\ddot{\theta}_2 \sin \gamma + g_L \\ \ddot{\psi}_{12} &= \frac{1}{l_{12}} \left(-\ddot{x}_2 \sin \phi + \ddot{y}_2 \cos \phi + d\ddot{\theta}_2 \cos \gamma + g_\psi \right) \end{aligned} \quad (3.7)$$

where for simplification $\phi = (\theta_1 + \psi_{12})$, and $\gamma = (\theta_1 + \psi_{12} - \theta_2)$ are used, and:

$$\begin{aligned} g_l &= -\ddot{y}_1 \sin \phi - \ddot{x}_1 \cos \phi + l_{12}(\dot{\theta}_1 + \dot{\psi}_{12})^2 - d\dot{\theta}_2^2 \cos \gamma \\ g_\psi &= \ddot{x}_1 \sin \phi - \ddot{y}_1 \cos \phi + d\dot{\theta}_2^2 \sin \gamma - 2\dot{l}_{12}(\dot{\theta}_1 + \dot{\psi}_{12}) - l_{12}\ddot{\theta}_1 \end{aligned} \quad (3.8)$$

The equations of motion of the boat (Eq. 3.4) are further simplified to give:

$$\begin{aligned} \ddot{x} &= \frac{\cos \theta}{m_{11}} [A + X] - \frac{\sin \theta}{m_{22}} [B + Y] \\ \ddot{y} &= \frac{\sin \theta}{m_{11}} [A + X] + \frac{\cos \theta}{m_{22}} [B + Y] \\ \ddot{\theta} &= \frac{1}{m_{33}} [C + N] \end{aligned} \quad (3.9)$$

where:

$$\begin{aligned} A &= (m_{22} - m_{11})v\dot{\theta} - d_{11}u \\ B &= (m_{22} - m_{11})u\dot{\theta} - d_{22}v \\ C &= -(m_{22} - m_{11})uv - d_{33}\dot{\theta} \end{aligned}$$

These simplified equations of motion are then inserted into the second derivative boat relation equations (Eq. 3.7). Thus yielding:

$$\begin{aligned}
\ddot{l}_{12} &= \left[\left(\frac{\sin \theta_2}{m_{11}} [A + X] + \frac{\cos \theta_2}{m_{22}} [B + Y] \right) \sin \phi \right. \\
&\quad \left. + \left(\frac{\cos \theta_2}{m_{11}} [A + X] - \frac{\sin \theta_2}{m - 22} [B + Y] \right) \cos \phi + d \left(\frac{1}{m - 33} [C + N] \right) \sin \gamma + g_l \right] \\
\ddot{\psi}_{12} &= \frac{1}{l_{12}} \left[- \left(\frac{\cos \theta_2}{m_{11}} [A + X] - \frac{\sin \theta_2}{m - 22} [B + Y] \right) \sin \phi \right. \\
&\quad \left. + \left(\frac{\sin \theta_2}{m_{11}} [A + X] + \frac{\cos \theta_2}{m_{22}} [B + Y] \right) \cos \phi + \frac{d}{m_{33}} [C + N] \cos \gamma + g_\psi \right]
\end{aligned} \tag{3.10}$$

To get the equations into the representation format normally used for control theory, for example $\dot{\mathbf{x}} = \mathbf{bu} + \mathbf{f}$, the relations are rewritten in matrix form:

$$\begin{bmatrix} \ddot{l}_{12} \\ \ddot{\psi}_{12} \end{bmatrix} = \begin{bmatrix} \frac{\cos \gamma}{m_{11}} & \frac{\sin \gamma}{m_{22}} & \frac{d \sin \gamma}{m_{33}} \\ \frac{-\sin \gamma}{m_{11} l_{12}} & \frac{\cos \gamma}{m_{22} l_{12}} & \frac{d \cos \gamma}{m_{33} l_{12}} \end{bmatrix} \begin{bmatrix} A + X \\ B + Y \\ C + N \end{bmatrix} + \begin{bmatrix} g_l \\ \frac{g_\psi}{l_{12}} \end{bmatrix} \tag{3.11}$$

This matrix equation is then further formatted by separating the loading terms X, Y, N from the others terms A, B, C and reducing the loading into just the X and Y forces using the relation $N = -LY$. These changes give:

$$\begin{bmatrix} \ddot{l}_{12} \\ \ddot{\psi}_{12} \end{bmatrix} = \begin{bmatrix} \frac{\cos \gamma}{m_{11}} & \sin \gamma \left(\frac{1}{m_{22}} - \frac{ld}{m_{33}} \right) \\ \frac{-\sin \gamma}{m_{11} l_{12}} & \frac{\cos \gamma}{l_{12}} \left(\frac{1}{m_{22}} - \frac{ld}{m_{33}} \right) \end{bmatrix} \begin{bmatrix} X \\ Y \end{bmatrix} + \mathbf{f} \tag{3.12}$$

or

$$\ddot{\mathbf{z}} = \hat{\mathbf{b}}\mathbf{u} + \hat{\mathbf{f}} \tag{3.13}$$

where f is:

$$\begin{bmatrix} \frac{\cos \gamma}{m_{11}} & \frac{\sin \gamma}{m_{22}} & \frac{d \sin \gamma}{m_{33}} \\ \frac{-\sin \gamma}{m_{11}l_{12}} & \frac{\cos \gamma}{m_{22}l_{12}} & \frac{d \cos \gamma}{m_{33}l_{12}} \end{bmatrix} \begin{bmatrix} A \\ B \\ C \end{bmatrix} + \begin{bmatrix} g_l \\ \frac{g_\psi}{l_{12}} \end{bmatrix}.$$

Recall that the loading terms are described at the beginning of this chapter, and thus the relation of the inputs n and α to the outputs l and ψ are obtained. It can be noted that this control law is similar to the preliminary one with the sole difference being changes to the \mathbf{f} , \mathbf{b} , and \mathbf{u} terms. Additionally, as with the original controller, the \mathbf{b} matrix is inverted and thus, invertibility is a requirement. To satisfy this $\det(\hat{\mathbf{b}}) = 0$ must be avoided. This simplified to a requirement of:

$$\frac{1}{m_{22}} - \frac{ld}{m_{33}} \leq 0 \quad (3.14)$$

From this we can satisfy the invertibility of \mathbf{b} and the stability of the controller by our choice of d , the distance from the boats CG to the control point.

3.3 Simulation and Discussion

The following section regards the simulation of the controller and boat plant. The reason for simulation is that it is easier to perform rapid prototyping and controller tuning with a plant model than on an actual boat. For the simulation of the boat plant and controller three path options are developed for the virtual lead boat. This was to simulate a wide range of system behaviors. The three lead paths chosen are the same as for the simulation of the initial boat controller, namely a straight line path, a circular path, and a sinusoidal shaped path. The following sections cover the equations that describe the lead paths and the considerations regarding the choice of the parameters for these paths, as these leader boat trajectories are different from the lead paths of the preliminary design. Lastly the results obtained from the simulations of the system for the three cases are shown and discussed.

3.3.1 Lead Boat Paths

There are three cases for the lead boat path. For each of these cases the paths described have to exhibit smoothness in position, velocity, and acceleration changes. That is, the paths cannot have any jerk, or other discontinuities. Also for the three cases, the simulation stops after 60 seconds and the follower is programmed to drift to a stop. The initially planned lead boat paths had both a ramp up and ramp down phase. However, due to instability during the deceleration, or ramp down phase, this phase is removed. This problem is due to zero-dynamic stability. A more detailed description of this problem is covered further in the conclusion and future work section at the end of the thesis. As a work-around to this issue, in place of the ramp down, the follower boat had the propeller angle and speed reset to zero and the boat was left to drift to a stop.

3.3.1.1 Straight line trajectory

The first, and default case is a straight line at an arbitrary angle. The following equations define this leader path: The initial phase of the path starts with a smooth ramping up of the velocity until time t_1 , during this phase the path is defined as:

$$\begin{aligned} \ddot{q} &= \frac{V\pi}{2t_1} \sin\left(\frac{\pi t}{t_1}\right) \\ \dot{q} &= -\frac{V}{2} \cos\left(\frac{\pi t}{t_1}\right) + \frac{V}{2} \\ q &= -\frac{Vt_1}{2\pi} \sin\left(\frac{\pi t}{t_1}\right) + \frac{Vt}{2} \end{aligned} \tag{3.15}$$

where q is a term describing the position along a straight line. This position q is later translated into global coordinates. The secondary phase, from t_1 to t_f has a constant

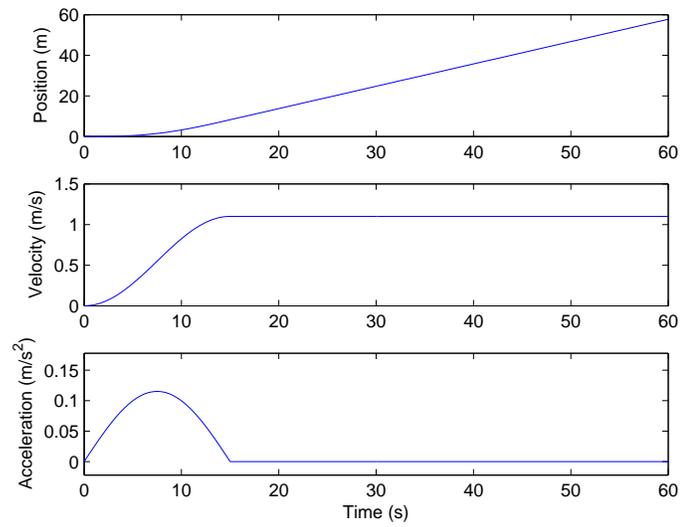
velocity profile as follows:

$$\begin{aligned}
 \ddot{q} &= 0 \\
 \dot{q} &= V \\
 q &= V(t - t_1) + \frac{Vt_1}{2}
 \end{aligned} \tag{3.16}$$

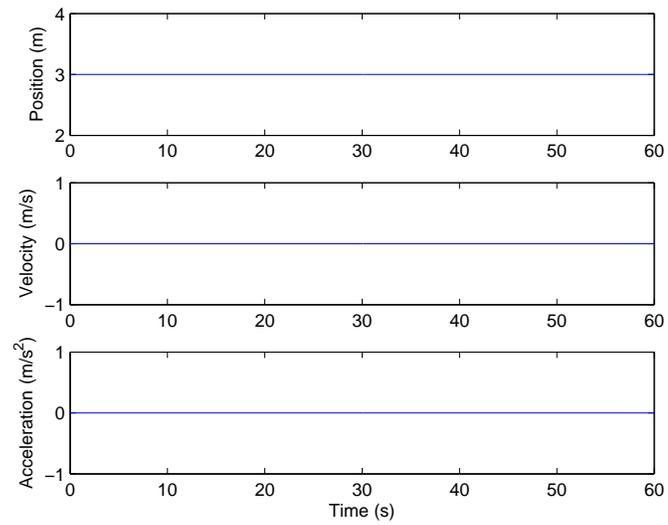
The identifying parameters for these equations are the times t_1 and t_f given as the time to ramp up and the finishing time. The ramping time is 15 seconds, and finishing time is 120 seconds. The V is the desired velocity along the path (here it is 1.1 m/s). From the path described by q , the global terms of the path are extracted through the relations:

$$\begin{aligned}
 \ddot{x}_1 &= \ddot{q} \cos(\alpha); & \ddot{y}_1 &= \ddot{q} \sin(\alpha); & \ddot{\theta}_1 &= 0; \\
 \dot{x}_1 &= \dot{q} \cos(\alpha); & \dot{y}_1 &= \dot{q} \sin(\alpha); & \dot{\theta}_1 &= 0; \\
 x_1 &= q \cos(\alpha) + x_{o1}; & y_1 &= q \sin(\alpha) + y_{o1}; & \theta_1 &= \alpha;
 \end{aligned}$$

Here α is the arbitrary angle of the path and y_{o1} , x_{o1} the starting position coordinates of the path. Note that the subscript “1” is to indicate the first case and that for this case the angle value used was $\alpha = 0$ radians. The following Fig. 3.4 and Fig. ?? show the x , y , and θ components of the lead path as well as the plot of the path on the cartesian plane for this case.

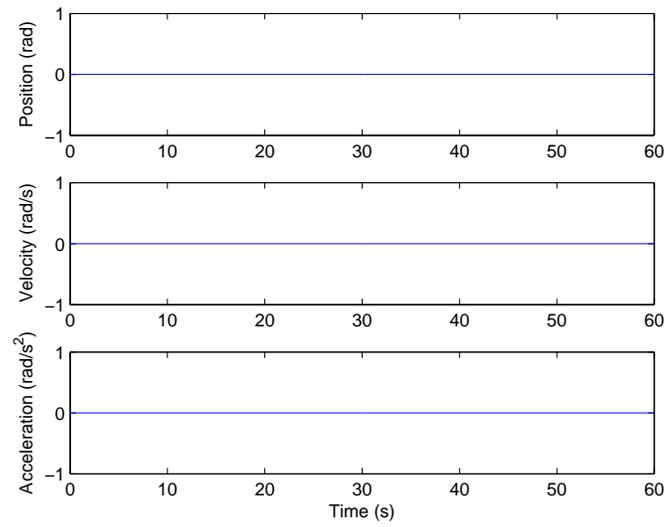


(a)

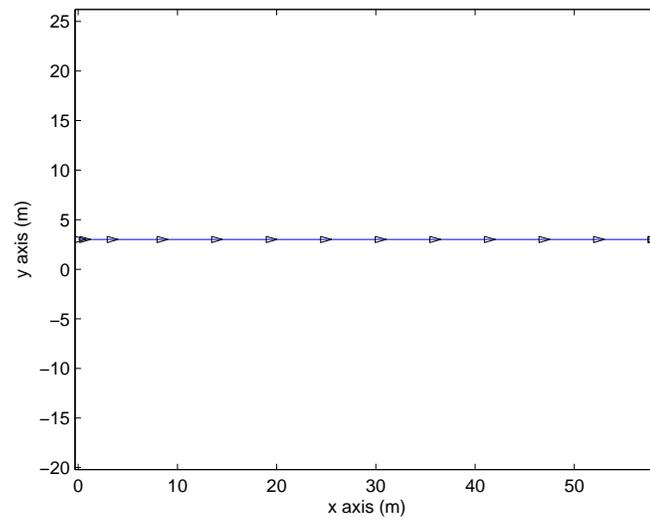


(b)

Figure 3.3: Straight Lead Boat: (a) x component; (b) y component.



(a)



(b)

Figure 3.4: Straight Lead Boat: (a) θ component; (b) Boat trajectory with orientation.

3.3.1.2 Circular trajectory

The circular path is similar to the straight path in terms of the ramp up and secondary phase. However, for this case the velocity is an angular one, and instead of a position q along the path, an angle ϕ is used. Thus, for the ramping up, or first phase, one can write:

$$\begin{aligned}\ddot{\phi} &= \frac{V_2\pi}{2t_1} \sin\left(\frac{\pi t}{t_1}\right) \\ \dot{\phi} &= -\frac{V_2}{2} \cos\left(\frac{\pi t}{t_1}\right) + \frac{V_2}{2} \\ \phi &= -\frac{V_2 t_1}{2\pi} \sin\left(\frac{\pi t}{t_1}\right) + \frac{V_2 t}{2}\end{aligned}\tag{3.17}$$

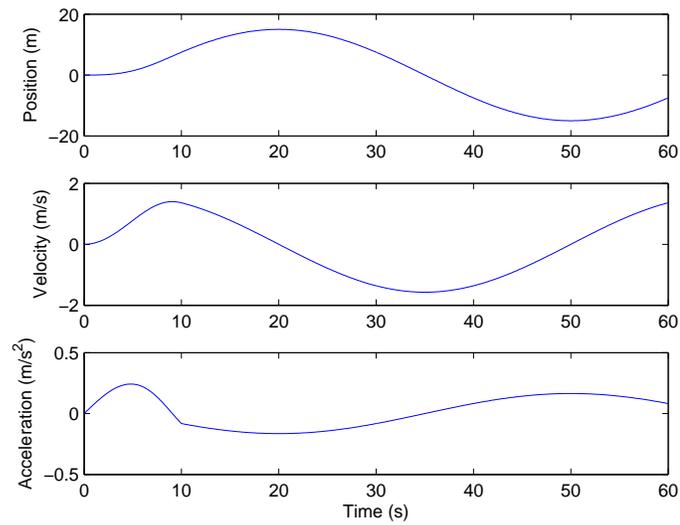
And for the secondary phase, the relations are:

$$\begin{aligned}\ddot{\phi} &= 0 \\ \dot{\phi} &= V_2 \\ \phi &= V_2(t - t_1) + \frac{V_2 t_1}{2}\end{aligned}\tag{3.18}$$

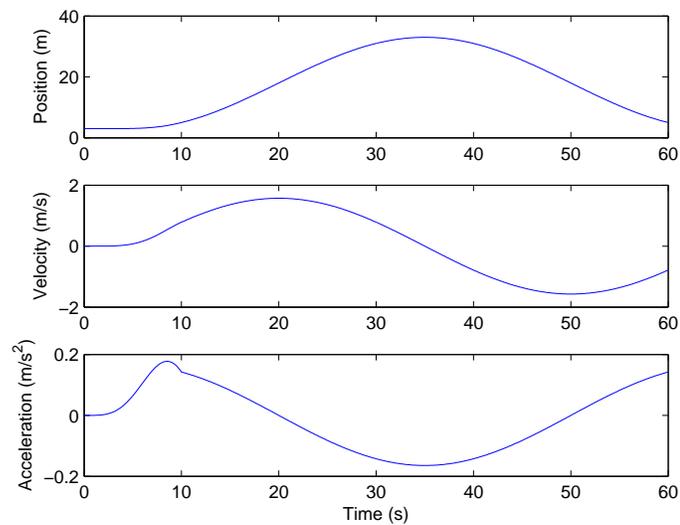
The main differences between this and the default path is that here V_2 is used, and it is an angular velocity. Here the velocity is ‘ $\pi/60$ ’ radians per second. The other differences are the times; for this path the ramp up period is only 5 seconds. The global terms are related to this ϕ value by the following relations, with the lemma that the ϕ includes the initial value of ϕ (ie. $\phi = \phi + \phi_o$):

$$\begin{aligned}\ddot{x}_2 &= -R \sin(\phi) \ddot{\phi} - R(\dot{\phi}^2) \cos(\phi) \\ \ddot{y}_2 &= R \cos(\phi) \ddot{\phi} - R(\dot{\phi}^2) \sin(\phi) \quad \ddot{\theta}_2 = \ddot{\phi}; \\ \dot{x}_2 &= -R \sin(\phi) \dot{\phi} \quad \dot{y}_2 = R \cos(\phi) \dot{\phi} \quad \dot{\theta}_2 = \dot{\phi}; \\ x_2 &= R \cos(\phi) + x_a \quad y_2 = R \sin(\phi) + y_a \quad \theta_2 = \phi + \pi/2\end{aligned}\tag{3.19}$$

Here R is the radius of the circular path (10 meters), and x_a, y_a are the coordinates of the center of the circle based on the initial position coordinates and the initial angle about the circle ϕ_o . They are defined as: $x_a = -R \cos(\phi_o) + x_{o2}$ and $y_a = -R \sin(\phi_o) + y_{o2}$. The initial positions and angle used are $x_{o2} = 0$, $y_{o2} = 2.9$, and $\phi_o = -\pi/2$. This path is depicted in Fig. 3.5 and 3.6:

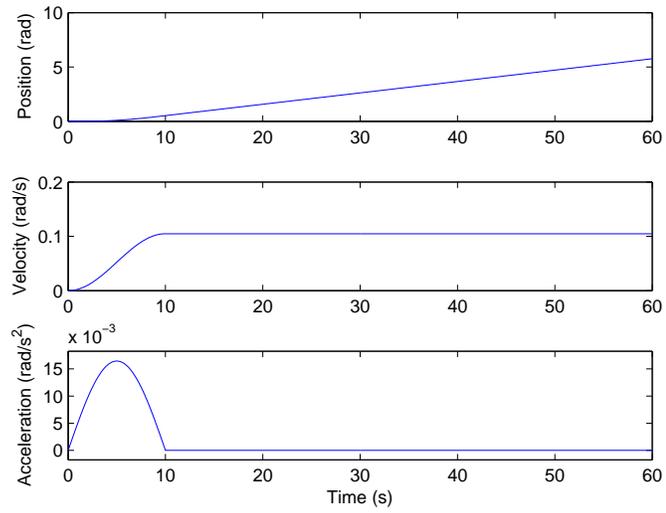


(a)

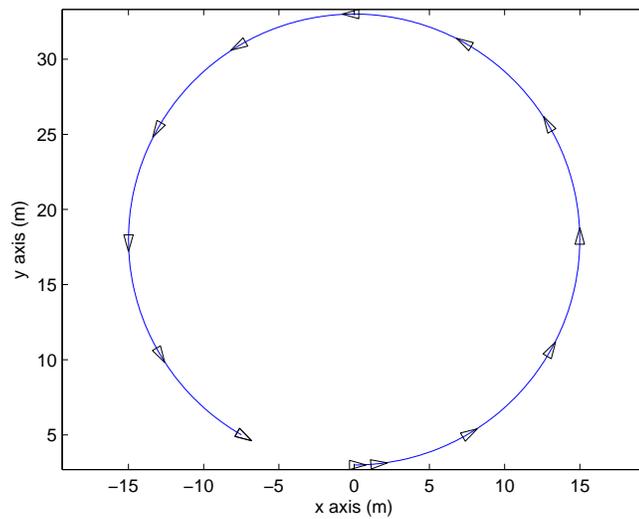


(b)

Figure 3.5: Circular Lead Boat: (a) Left: x component; (b) y component.



(a)



(b)

Figure 3.6: Circular Lead Boat: (a) θ component; (b) Boat trajectory with orientation.

3.3.1.3 Zigzag trajectory

The third case is the zigzag shaped path. For this case, the path is broken into sections similarly to the above paths except that instead of just a ramp up and constant phase, there are seven curved sections that combine to form a zigzag shaped path. The speed of the path along the x axis is defined the same as the above cases.

For the first phase:

$$\begin{aligned}\ddot{q} &= \frac{V_3\pi}{2t_1} \sin\left(\frac{\pi t}{t_1}\right) \\ \dot{q} &= -\frac{V_3}{2} \cos\left(\frac{\pi t}{t_1}\right) + \frac{V_3}{2} \\ q &= -\frac{V_3 t_1}{2\pi} \sin\left(\frac{\pi t}{t_1}\right) + \frac{V_3 t}{2}\end{aligned}\tag{3.20}$$

And for the secondary phase:

$$\begin{aligned}\ddot{q} &= 0 \\ \dot{q} &= V_3 \\ q &= V_3(t - t_1) + \frac{V_3 t_1}{2}\end{aligned}\tag{3.21}$$

Using these to get:

$$\begin{aligned}\ddot{x}_3 &= \ddot{q}; \\ \dot{x}_3 &= \dot{q}; \\ x_3 &= q + x_{o3};\end{aligned}$$

Note that the desired velocity V_3 for this case is one meter per second and x_{o3} is 0 meters. Using this baseline x axis velocity profile, the path curves are produced by defining the parameters of a seventh order polynomial for the y axis. These

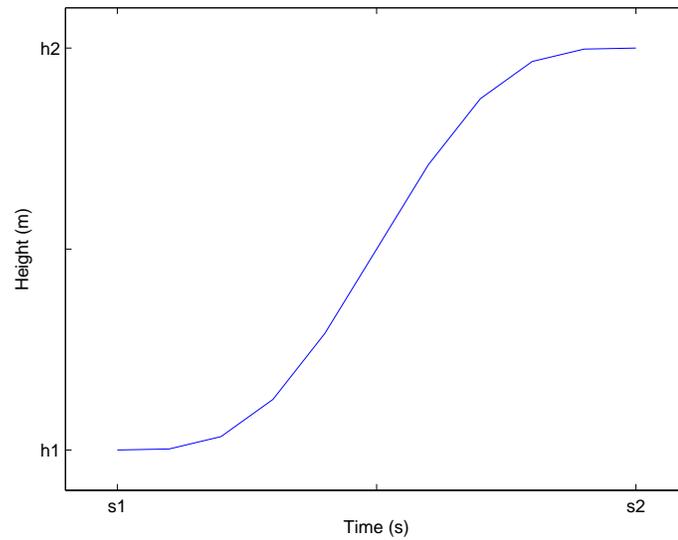


Figure 3.7: Example of curve with starting and ending times and positions

polynomials are seventh order so that there is continuous jerk, and are defined as:

$$\begin{aligned}
 y &= a_7t^7 + a_6t^6 + a_5t^5 + a_4t^4 + a_3t^3 + a_2t^2 + a_1t + a_0 \\
 \dot{y} &= 7a_7t^6 + 6a_6t^5 + 5a_5t^4 + 4a_4t^3 + 3a_3t^2 + 2a_2t + a_1 \\
 \ddot{y} &= 42a_7t^5 + 30a_6t^4 + 20a_5t^3 + 12a_4t^2 + 6a_3t + 2a_2 \\
 \dddot{y} &= 210a_7t^4 + 120a_6t^3 + 60a_5t^2 + 24a_4t + 6a_3
 \end{aligned} \tag{3.22}$$

The important parameters of the polynomial are: start time (s_1), start height (h_1), end time (s_2), and end height (h_2). Refer to Fig. 3.7 showing an example curve with these starting and ending times and positions.

To get a curve as shown in Fig. 3.7 the a_i terms from Eq. 3.22 are defined according to:

$$\begin{aligned}
a_0 &= (21s_2^5h_1s_1^2 - 7s_2^6s_1h_1 + 7h_2s_2s_1^6 - 35s_2^4s_1^3h_1 \\
&\quad + 35h_2s_2^3s_1^4 - 21h_2s_1^5s_2^2 - h_2s_1^7 + s_2^7h_1)/b \\
a_1 &= -140(-h_1 + h_2)s_2^3s_1^3/b \\
a_2 &= 210(-h_1 + h_2)s_2^2s_1^2(s_2 + s_1)/b \\
a_3 &= -140(-h_1 + h_2)s_2s_1(s_2^2 + 3s_2s_1 + s_1^2)/b \\
a_4 &= 35(s_2^3 + 9s_1s_2^2 + 9s_2s_1^2 + s_1^3)(-h_1 + h_2)/b \\
a_5 &= -84(-h_1 + h_2)(s_2^2 + 3s_2s_1 + s_1^2)/b \\
a_6 &= 70(s_2 + s_1)(-h_1 + h_2)/b \\
a_7 &= -(-20h_1 + 20h_2)/b
\end{aligned} \tag{3.23}$$

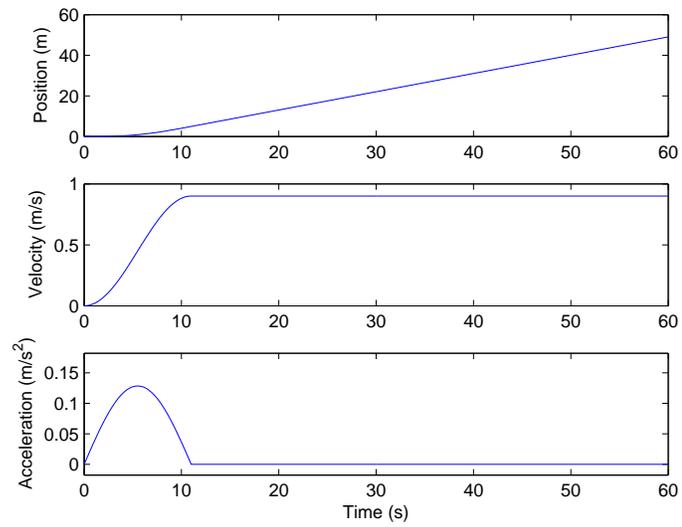
where b is: $(7s_2s_1^6 - 21s_1^5s_2^2 + 35s_2^3s_1^4 - 35s_2^4s_1^3 + 21s_1^2s_2^5 - 7s_1s_2^6 - s_1^7 + s_2^7)$. This polynomial is used seven times to produce seven curved segments. The combination of these seven segments creates the zigzag shaped path. Encasing the curved segments are two straight sections for the ramping up done for the x velocity profile and straightening out prior to finishing. To produce these curved and straight sections the start/end times and positions are defined for each segment as follows:

$$\left\{ \begin{array}{ll}
 s_1 = 0, s_2 = t_1, h_1 = 0, h_2 = 0 & \text{if } 0 < t \leq t_1 \\
 s_1 = t_1, s_2 = t_2, h_1 = 0, h_2 = h_{top} & \text{if } t_1 < t \leq t_2 \\
 s_1 = t_2, s_2 = t_3, h_1 = top, h_2 = h_{bot} & \text{if } t_2 < t \leq t_3 \\
 s_1 = t_3, s_2 = t_4, h_1 = bot, h_2 = h_{top} & \text{if } t_3 < t \leq t_4 \\
 s_1 = t_4, s_2 = t_5, h_1 = top, h_2 = h_{bot} & \text{if } t_4 < t \leq t_5 \\
 s_1 = t_5, s_2 = t_6, h_1 = bot, h_2 = h_{top} & \text{if } t_5 < t \leq t_6 \\
 s_1 = t_6, s_2 = t_7, h_1 = top, h_2 = h_{bot} & \text{if } t_6 < t \leq t_7 \\
 s_1 = t_7, s_2 = t_8, h_1 = bot, h_2 = 0 & \text{if } t_7 < t \leq t_8 \\
 s_1 = t_8, s_2 = t_f, h_1 = 0, h_2 = 0 & \text{if } t_8 < t \leq t_f
 \end{array} \right. \quad (3.24)$$

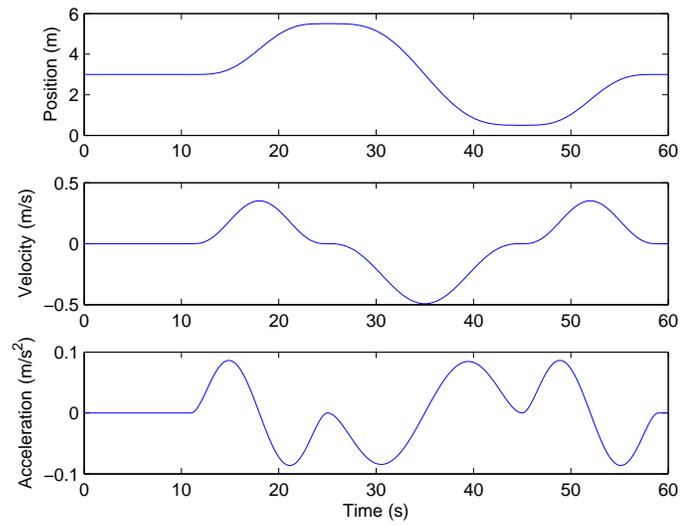
where the times t_i indicated above are: $t_1 = 5$ s, $t_2 = 20$ s, $t_3 = 35$ s, $t_4 = 50$ s, $t_5 = 65$ s, $t_6 = 80$ s, $t_7 = 95$ s, $t_8 = 110$ s, $t_f = 120$ seconds respectively, and the heights: $h_{top} = 2.5$ m and $h_{bot} = -2.5$ m. Combining the above, yields the desired y axis component of the path as follows:

$$\begin{aligned}
 \ddot{y}_3 &= \ddot{y}\dot{x}_3 & \ddot{\theta}_3 &= \ddot{y} \\
 \dot{y}_3 &= \dot{y}\dot{x}_3 & \dot{\theta}_3 &= \dot{y} \\
 y_3 &= y + y_{o3} & \theta_3 &= \dot{y}
 \end{aligned}$$

Here the initial y position y_{o3} is 3 meters. The resulting compilation path is shown in Fig. 3.8 and 3.9.

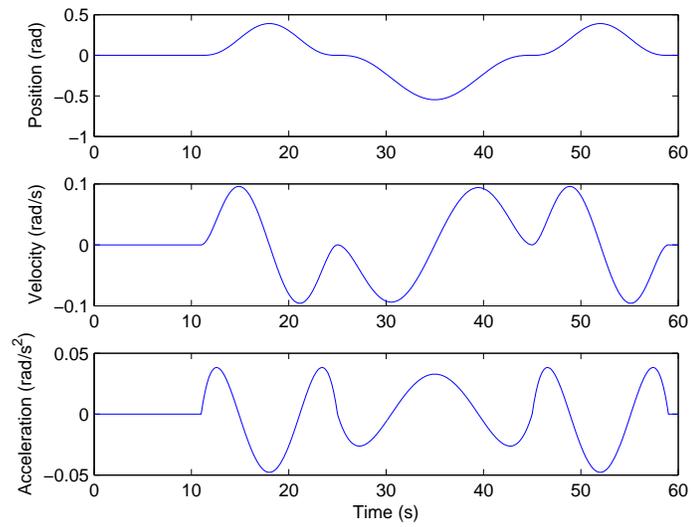


(a)

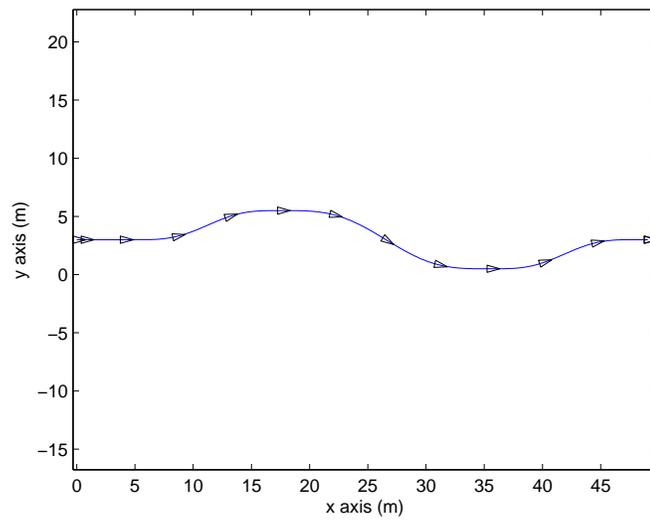


(b)

Figure 3.8: Zigzag Lead Boat: (a) x component; (b) y component.



(a)



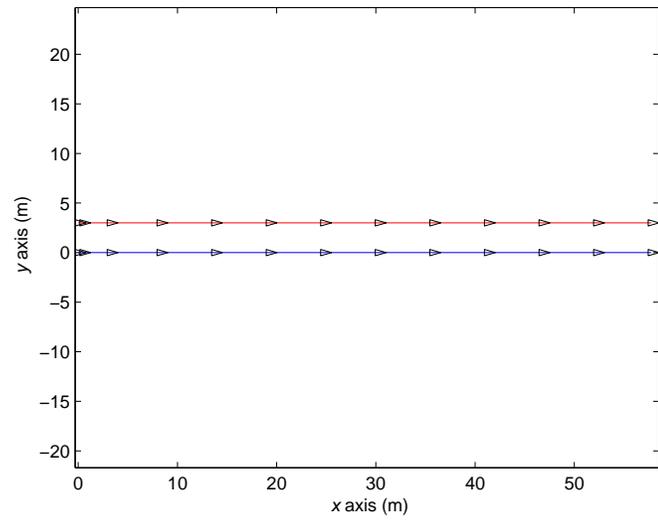
(b)

Figure 3.9: Zigzag Lead Boat: (a) θ component; (b) Boat trajectory with orientation.

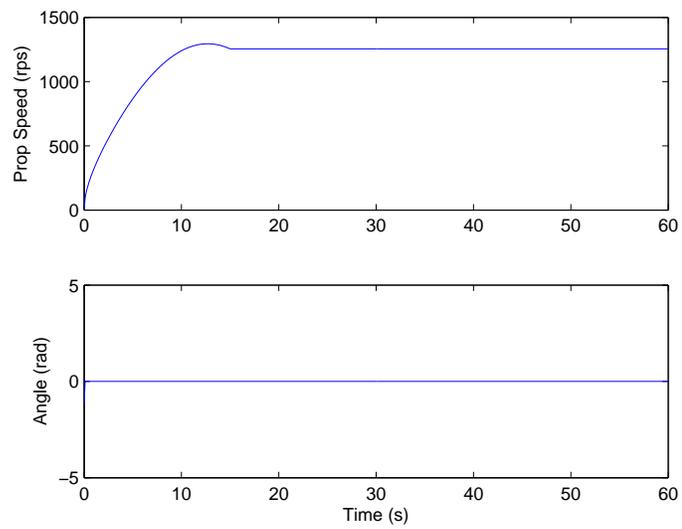
3.3.2 Simulation results

The lead boat paths are generated as one of the inputs to the control system (along with the desired values). With these inputs and a model of the actual boat (the plant) and the control laws (the controller) the system becomes a closed loop system. Simulation, of this system as a whole, is done and the following figures (Fig. 3.10 to 3.14) show the results. Note that each lead path trajectory is simulated as a separate case.

The first case is the one with the lead boat trajectory following a straight line path. The initial conditions of the follower boat for this case are: x , y positions both zero, angle θ also zero, and boat is at rest so velocities are all zero as well. Fig. 3.10, 3.11 shows the paths of the leader and follower boat in the cartesian plane, the input to the follower boat, and the outputs of the system, respectively.



(a)



(b)

Figure 3.10: Case 1: Simulation of straight line path (a) Boat trajectories with orientations; (b) Follower inputs.

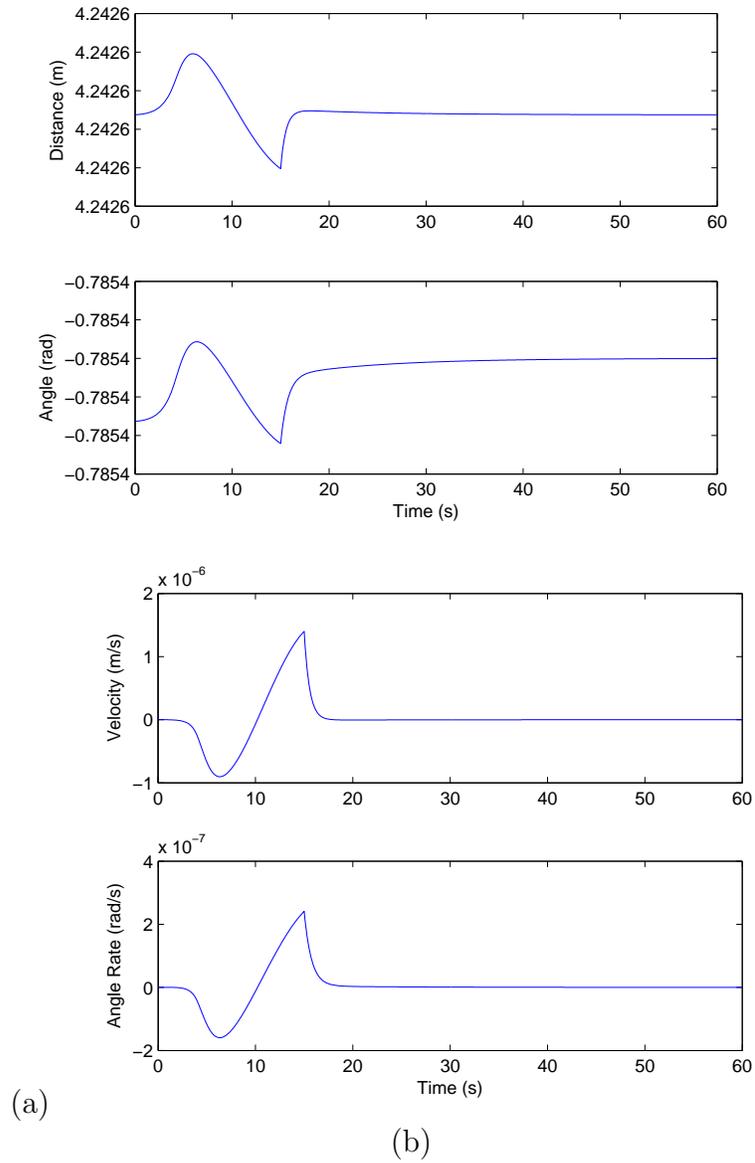
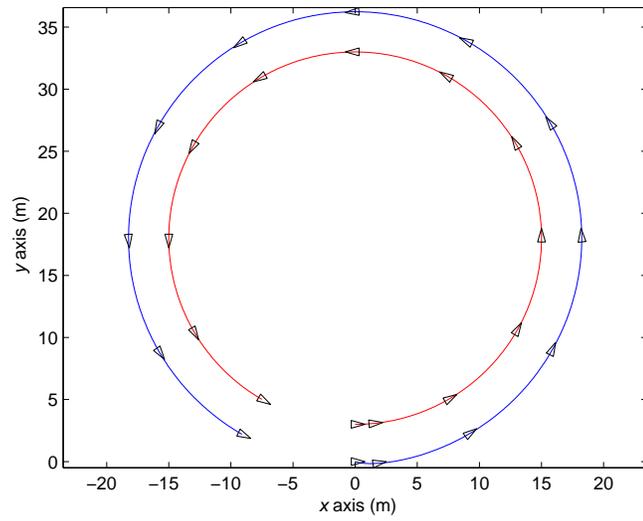
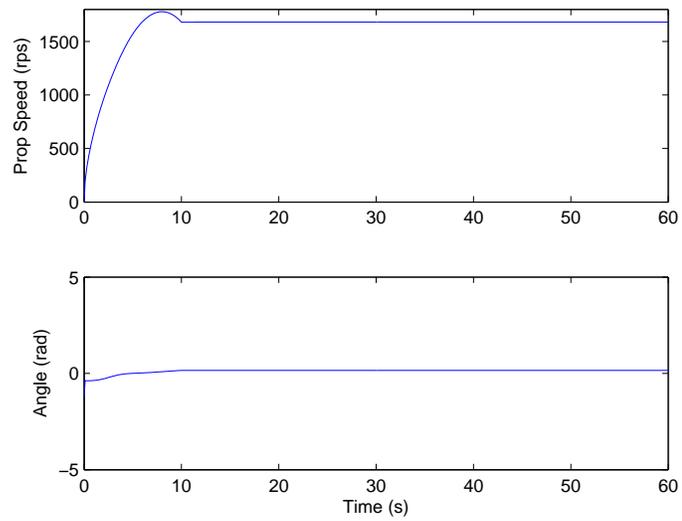


Figure 3.11: Case 1: Simulation of straight line path (a) System outputs; (b) Derivatives.

The second case is the circular lead path. The initial conditions of the follower boat for this case are the same as in the previous case. All positions (x, y, θ) and velocities $(\dot{x}, \dot{y}, \dot{\theta})$ are zero. Fig. 3.12, 5.12 shows the paths of the leader and follower boat in the cartesian plane, the inputs to the follower boat, and the outputs of the system as the previous case.

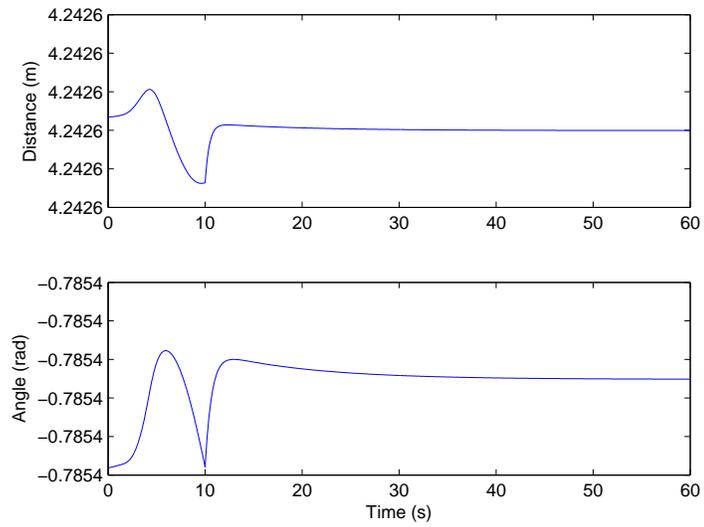


(a)

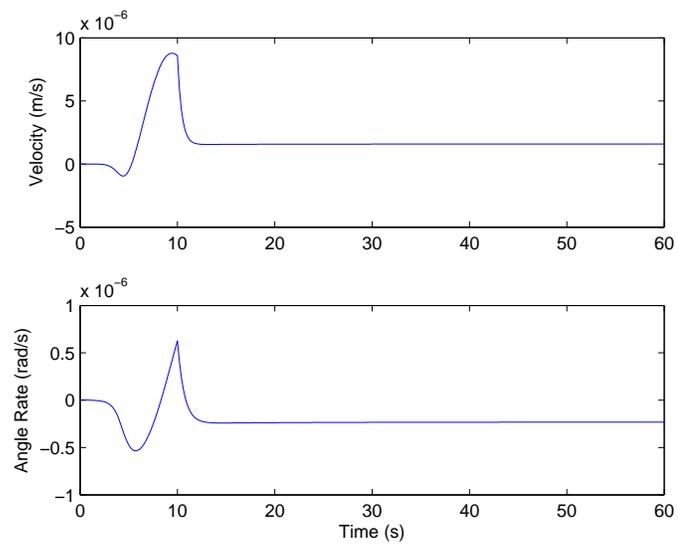


(b)

Figure 3.12: Case 2: Simulation of circular path (a) Boat trajectories with orientations; (b) Follower inputs.

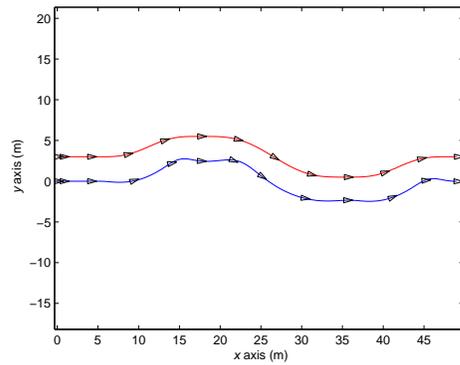


(a)

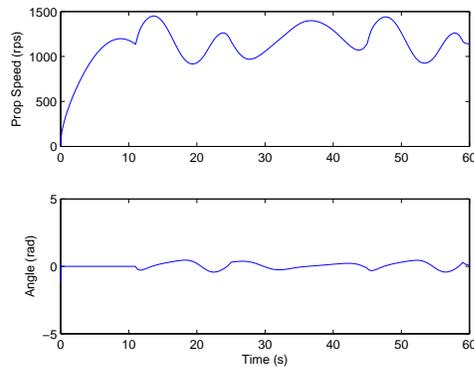


(b)

Figure 3.13: Case 2: Simulation of circular path (a) System outputs; (b) Derivatives.



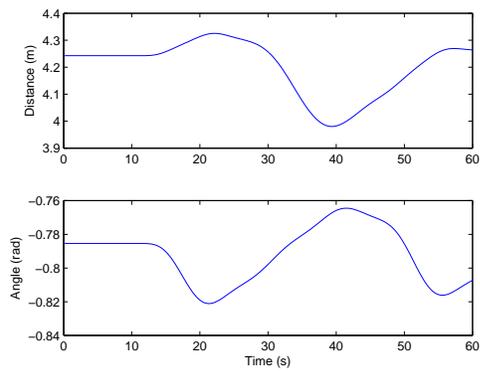
(a)



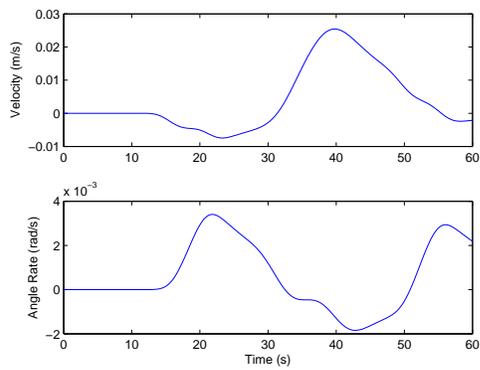
(b)

Figure 3.14: Case 3: Simulation of zigzag shaped path (a) Boat trajectories with orientations; (b) Follower inputs.

The third and final case is the zigzag shaped path. The initial conditions of the follower boat for this case are the same as in the previous two cases. All positions (x, y, θ) and velocities $(\dot{x}, \dot{y}, \dot{\theta})$ are zero. Fig. 3.14 and 3.15 shows results of this simulation case.



(a)



(b)

Figure 3.15: Case 3: Simulation of zigzag shaped path (a) System outputs; (b) Derivatives.

3.3.3 Discussion

From these simulation results it is discovered at which point the system would become unstable. For example, as mentioned earlier regarding the zero-dynamic stability, when the vessel decelerated for the circular path simulations it would sometimes go into a spinning loop at the end instead of slowing to a stop as desired. This, and other cases, are due to the controller setting the control point to have zero velocity, but not the boats CG. This is rectified by having the control point be able to shift from in front, to behind the vessel. However, this results in controller instabilities, so the simple solution is to forbid, or constrain, the lead boat paths such that these situations would not arrive. This same technique is used for the zigzag and straight case to determine the scope of lead boat trajectories that would produce viable follower/controller responses, ones that are stable for the whole period and have minimal error.

With this robust, within set leader constraints, controller and model the next step is implementation of the model onto the physical vehicle. But before this can happen the hardware needs to be constructed and assembled to provide the scaffold for the formation controller.

CHAPTER 4

EXPERIMENTAL SETUP

The following chapter covers the hardware used for the implementation of the formation control of a marine vehicle. The setup and configuration of the major components are described as well. Further, the enmeshing of the components and the supporting containment and structures are detailed.

4.1 Sensor

The Sensor used was an AHPRS from Rotomotion. This unit is a sensor array that provides the position, velocity, acceleration, angular orientation, rotational velocities, magnetometer, and voltages of the unit. The original interface for this system is a parallel port that contained power lines as well as serial and Ethernet communication lines. Only the Ethernet line is used for data transfer, thus making it easy to wire/interface to other components. The serial is for setting the systems IP address and other system parameters via a telnet program and is not used. More information about these parameters and the system can be found in the data sheet “DOC-AHRS24-REFMAN-001C.pdf” [22].

4.1.1 Setup

The system requires a power supply with voltage of at least 7.2 V DC, and can take unregulated power up to 14 V as there are two internal voltage regulators that limit it to the proper internal voltages for each sensor component. If the power supply is wired backwards these regulators will overload and need to be replaced. This may also cause the sensor to require recalibration.

Use of the AHPRS required first supplying power then use of a network cable to connect the system with a computer. Note that the wiring is such that a crossover cable is not required. Secondly, the IP address of the network adaptor of the computer should be set to 10.0.112.101 (or higher) to make it on the same network and subnet as the sensor, whose IP address is set to 10.0.112.100. To do this in Windows OS open network connections, choose the network adaptor and right click, selecting properties. On the first tab in the properties window select internet protocol (TCP/IP), click properties button. In the new window manually enter the IP address and subnet for this network. Close the window, and on the properties window select the “advanced” tab. Disable the firewall and close the window. It should now be possible to connect to the AHPRS.

4.1.2 Configuration and Calibration

Finally, using a software program such as ground.exe, or “AHRSviewer” (not used as it is included in “ground” program) both of which are found on the Rotomotion software disk, or a MATLAB Simulink model, communicate with the sensor to get the data stream started and begin reading out the dynamic states. To communicate, the software needs to send data to IP 10.0.112.100 and port 2002, the current parameters of the AHPRS. More details about the software will come later in this chapter. A simple test to check that the sensor is working is to use the ground program with

the following command in the command line: “ground.exe -s 10.0.112.100 -l logfile” from the appropriate directory. The way to check for a working sensor with these programs is twofold. First in the ground program the status tab will show the current values of angles, accelerations, etc. It will also have a counter in the bottom right corner showing the number of packets received. The second method is to use a log file to record the data, then check the state values in the log file comparing them to the actual motion of the AHPRS.

The calibration and recalibration of the sensor is done by Rotomotion staff at their manufacturing facility. The only sensor component that can be recalibrated by the user is the magnetometer. This can be done with only a compass and a computer running the ground program. The procedure is described in the manual [22].

4.2 Drive Train Control: SSC Configuration and Setup

The servo switch/controller (SSC) is used as a switching bus to connect either the radio control receiver or the serial port of the onboard computer to the boat servo and speed regulator. The SSC receives a command signal from the RC receiver and using this signal sets the outputs based on a lookup table [23]. To set up the SSC, it must first be connected with a serial adaptor to a PC running HyperTerminal, then power cycled. Upon restart, a short message will appear, indicating the user to enter “ssconfig” to enter setup mode. Doing this allows the user to reconfigure the switch table and other option of the flash ROM of the SSC. For the boat, since only two outputs are needed the first and second channels are set to correspond to the first and second channels of the RC receiver for the primary command signal (1 Hz), and to correlate to the serial port for the secondary command signal (2 Hz). For the default case (receiver or transmitter is off) the outputs are set to a constant value, they are also set to this for the intermediate command signal (1.5 Hz). The SSC outputs are

all set to have a broadcast frequency of zero (off) except for the status packet, set to 10 HZ, allowing the PC/104 to know in which command signal, or state, the SSC is.

To send and receive packets between the SSC and the onboard computer Simulink blocks are made. These blocks utilized the core serial interface blocks (RS232 Binary Receive, and Binary Send) with additional subfunctions to extract the useful bits from the binary packet for receiving. For the case of sending the serial packet to the SSC, to pack the binary packet with the input values and corresponding checksums. When sending the packets to the SSC, the important values are the pulse widths of the signal for each channel. As mentioned earlier, as there are only a rudder servo and speed controller, only two channels are needed and the rest of the channels are sent a null value of 1500 ms. This pulse width equates to an orientation of zero radians for the servo, or a speed of zero for the speed regulator.

To compute the required pulse width for each desired angular position or propeller speed some testing needs to be done. The rudder angle is found to be limited to within $\pm \frac{\pi}{6}$ radians and this corresponds with pulses between 1000 to 2000 ms. Similar testing for the propeller speed relation yields a nonlinear relation. The relation equations determined from this testing are:

$$pw_A = s_A \frac{1000}{\pi} + 1500 \quad (4.1)$$

$$pw_S = -0.0002s_{Sp}^3 + 0.03s_{Sp}^2 - 2.66s_{Sp} + 1528 \quad (4.2)$$

where pw_A is the angle pulse width and pw_S is the speed pulse width and s_A and s_{Sp} are the desired servo angle and speed, respectively.

4.3 Onboard Computer

The onboard computer runs the servos through the SSC and couples the sensor unit with the outputs of the vehicle. The data sheet describing the wiring of the

PC/104 [24], also describes the form factor structure of the computer and how additional layers (e.g. network adaptor) can be added. The computer runs DOS, and based on this is the XPCtarget kernel from MATLAB. This kernel runs the Simulink models on the target PC and allows some network communication such as FTP with the hard drive of the target from a host PC running MATLAB. Refer to MATLAB help files and website for more detailed information about this under “xpcexplr”.

For the setup of the computer the only thing of note is that the kernel must be initialized to use TCP communication with a host to enable any and all other network communication. If serial is chosen for host-target communication then the PC/104 will not be able to communicate to the AHPRS. Other cautions: The heat sink of the PC/104 gets very hot; be careful about touching it after the PC has been running for extended periods. Also, the pins used to connect the layers of the PC/104 can get bent easily when assembling/disassembling/rearranging layers. Thirdly, use care when connecting and unplugging the hard drive from its IDE cable as its pins get bent easily as well.

To initially run the PC/104 a DOS environment will have to be copied to the hard drive, using a USB drive mate to connect the HDD to another computer (this is not verified, it may be possible to run the programs directly without DOS). This method was also used to write standalone programs (Simulink models) to the HDD, and the xpcboot, and xpctarget kernel programs. Refer to MATLAB help about these.

The software run on/by the xpctarget kernel is covered in more detail in the preceding sections about the theoretical models and simulations of the boat and controller, and also the intermediate software to connect the controller and open loop simulations to the sensor and SSC will be covered in chapter 5.

4.4 Controller Box

To lower the size and weight of the electronic components the AHPRS and the PC/104 are repackaged into a single container along with a local power supply (Li-Po battery) and the SSC. An internal framework for mounting the PC/104 and sensor components together is designed and switches to toggle power to the computer and sensor are added to the exterior of the box along with ports for the GPS antenna connector, servo cabling, and RC antenna. Detailed information about this box form factor and wiring can be found in the Appendix B.

4.5 Boat Hull Reconfiguration

To fully utilize the Traxxas model boat for testing the formation control scheme some modifications and additions to the boat are required. The primary addition is to mount the sensor box to the boat. Since the box is too large to fit into the hull of the boat it is mounted to the top, causing the center of gravity to move up to the point where the boat is unstable about its longitudinal axis. Thus a second addition is required, the pontoons and structural spars.

4.5.1 Design

Prior to designing the box mount and pontoons, the components of the ensemble are weighed and the boat hull was measured out. Refer to Fig. 4.1. Along with this, the density of the pontoon material is determined. Note that the pontoon material for this first prototype was an aged piece of balsa timber. Using this information (table 4.1), calculations are performed to confirm that the boat would not sink and to find how much of the pontoons would need to be submerged, to provide the proper amount of buoyancy. This determines the depth of the pontoons, with respect to the boat hull, as a design parameter.

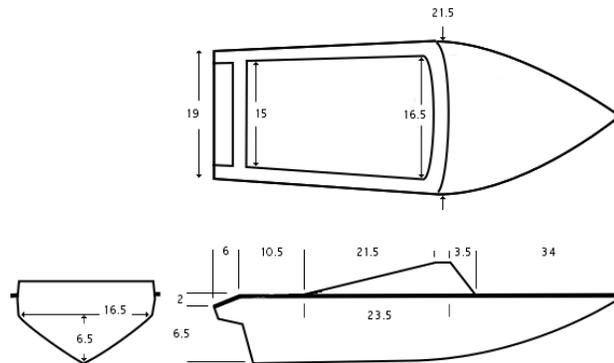


Figure 4.1: Rough Dimensions of Boat (in centimeters)

Table 4.1: Weights of Components

1500mAh battery	292 g
4200mAh battery	431 g
Sensor box w receiver	2827 g
Receiver	38 g
Boat	1506 g
Density of Balsa	0.1254 g/cm ³

Other design parameters that need to be considered included the fact that the sensor box has to be level and that the box should be able to be shifted forward and backwards to adjust the center of gravity (CG) of the system. Similarly, the addition of the box mount and the pontoons should not shift the CG of the boat either. To this affect the box mount is kept as low as possible on the boat, and the whole addition has bilateral symmetry.

Several methods of mounting to the boat were considered but the final design uses two grooved wedges that fit over the sidewalls of the boat to which a flat wood plank is fastened. This plank then acts as a mounting platform for the sensor box and the spars for connecting the pontoons to the boat. The grooves in the wedges prevent



Figure 4.2: The Boat Mount Platform

any lateral motion of the mounting platform, and the platform itself uses four clamps to hold the platform down tight against the boat (these clamps hook under the lip of the boat hull). Additionally, due to their shape and the clamps, the wedges prevent the platform from moving forwards and a small waterproofed shim is used between the back of the wedges and the watertight container protecting the servo and speed controller, to prevent the platform from moving backwards with respect to the boat. Refer to Fig. 4.2 showing the boat mount platform.

The design of the shape of the pontoons is such that it would be the simplest shape to construct and that removed the least amount of material from the balsa timbers. The shape chosen has 45 degree angles for the bottom of the pontoon and

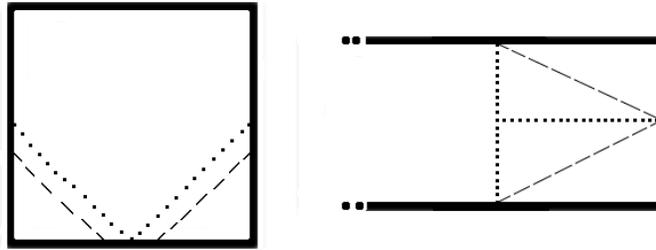


Figure 4.3: Diagram of Cutting lines for Shaping Pontoons

1:2 ratio for the angle of the prow. Refer to Fig. 4.3 showing the cutting outline used for construction.

4.5.2 Fabrication

The pontoons are constructed from a single balsa timber. The piece is cut in half lengthwise to yield two sections approximately 59 cm in length and a cross section of 9 by 10 cm. The pieces are marked with lines to indicate where the 45 degree section is. Lines, marking where to cut, are then put approximately 1 cm further out, thus giving room for sanding down the corners and giving the pontoon hulls a smoother rounder shape. A similar procedure is followed for the prows of the pontoons except that the cutting lines were marked exactly along the 1:2 angle lines. See Fig. 4.3. The corners from these cuts are then sanded down as before. Once the rough sanding is completed a fine grain sand-paper was used, sanding along the grain of the wood, to give it a nice finish in preparation for lacquering. Also prior to the lacquering, strips of wood are glued to the top surface to provide a better anchor for the screws that will later fasten the pontoons to the spars. Refer to Fig. 4.4 showing the pontoons and connecting spars.

The boat platform is constructed in the following manner: First the wedges are made by gluing together three different shaped wedges. This is to produce the groove in the wedge that fits over the side wall of the boat. These wedges then have some

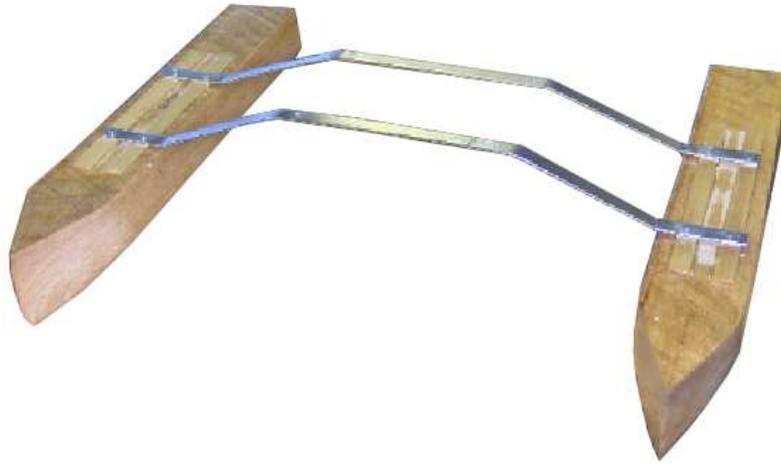


Figure 4.4: The Pontoons and Connecting Spars

corners sanded to better fit to the corners of the boat mold. With the wedges in place on the boat the platform plank is placed on top and the locations of the wedges are marked. The wedges are then glued to the platform in the marked locations and holes are drilled through the platform, around the wedges, for the clamps. Four slots are also made into the prow (two), and stern (two), edges of the platform for fastening the sensor box to the platform. The platform plank is then trimmed and sanded, to comply with the bilateral symmetry and in preparation for lacquering, respectively. Refer to Fig. 4.2.

The lacquering is done in a single day, according to the application instructions, giving the pontoons a triple coat since they are to be partially submerged, and the boat mount a double coat. After 24 hours of curing the parts are sanded a final time.

The spars that connect the pontoons to the boat mount platform are strips of aluminum flat bar. Mounting holes are drilled such that they fit on the same bolts used to clamp the platform to the boat, and on each end for the screws that fasten the pontoons to the spars. The spars are bent to set the relative height of the pontoons to the boat hull. Refer to Fig. 4.4 showing the spars fastened to the pontoons.

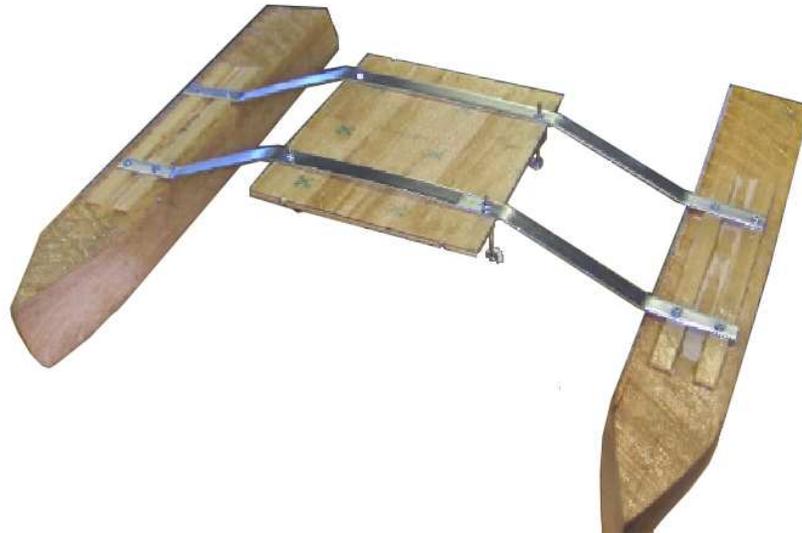


Figure 4.5: The Boat Mount Connected to pontoons via Spars

The boat mount platform when connected to the spars is shown in Fig. 4.5 and Fig. 4.6 shows a close up of the clamps that keep the boat mount snug against the boat hull. The complete assembly fastened to the boat is shown in Fig. 4.7



Figure 4.6: Close view of Clamps used to Fasten Boat Mount to the Hull of the Boat



Figure 4.7: The Complete Assembly on the Boat

CHAPTER 5

EXPERIMENTAL RESULTS - SIMPLE MODEL

With the hardware and the simulation and modelling software developed, the final step is implementation. To do this the hardware and software aspects are integrated into a controller interface linking sensor and software programs. Upon completion, testing and experiments are conducted to parameterize the vessel and to test the controller algorithms.

5.1 Integration of subsystems and real-time control program

In order to engage the computer with the RC trigger mechanism, the computer needs to be able to tell the status of the SSC. This was done by utilizing an output message option of the SSC to send a packet digitally describing the outputs. The status message (refer to section 4.2.1.1 in SSC documentation [23]) indicates the value of the command signal, and thus, indicates if the system is triggered or not. To have the PC/104 computer read this message, a serial connection is established between the PC/104 and the SSC. Thus, using COM port (serial port) blocks in Simulink, the PC/104 computer is able to both send and receive packets with the SSC. The sending is also important as this enables the controller to send out the pulses to control the steering servo and motor speed controller. An additional subsystem is introduced to convert the controller outputs into the correct packet structure as required by the

SSC serial connection.

To get the state of the boat as an input, the sensor values need to be read by the PC/104 computer. This is done in a similar fashion to the SSC to PC interface, however, for the sensor, the connection type is User Datagram Protocol (UDP) over ethernet crossover cable. As with the SSC, UDP send and receive blocks are used in the Simulink model to transmit and receive the UDP packets. Additionally, pack and unpack blocks are used to formulate the data into the correct packet structure. On startup of the system, there is a thirty second delay to wait for both the sensor and other components to finish their respective startups, then an initialization packet is sent from the PC/104 computer to the sensor. This starts the sensors broadcast of the measured values back to the IP address of the PC/104 computer, at the port designated by the initialization packet. Thus the UDP receive block is configured to listen for packets from the IP of the sensor and at the same port as designated by the UDP send block. Furthermore, within the sensor reader subsystem, the UDP packets from the sensor are filtered to output only the state packets and then extract them to stream out the values for each component.

The positions, as output from the sensor reader subsystem, are reset for the triggering of the controller to prevent the system from having to deal with large jumps and errors for start conditions. Thus the sensor reader outputs are post processed to reset the position values to zero when the system is triggered. Additionally, the post processing function also projects the values onto the horizontal plane to remove any velocity and acceleration components introduced by the pitch and roll of the boat, as well as shifting the values of the state vector to include the global coordinates. This is to match the state vector that was used in the simulations $(x, y, \theta, \dot{x}, \dot{y}, \dot{\theta})$ so that an identical controller function can be used. Note that the controller uses only local values (u, v, r) and thus the global values are converted back, within the controller, as with the simulation version, however the global values are used in the calculation

of the system outputs l_{12} and ψ_{12} .

The sensor reader, sensor post processing, SSC status detection, and serial output to SSC subsystems are combined together with the controller subsystem to complete the onboard system. These components are shown in Fig. 5.1

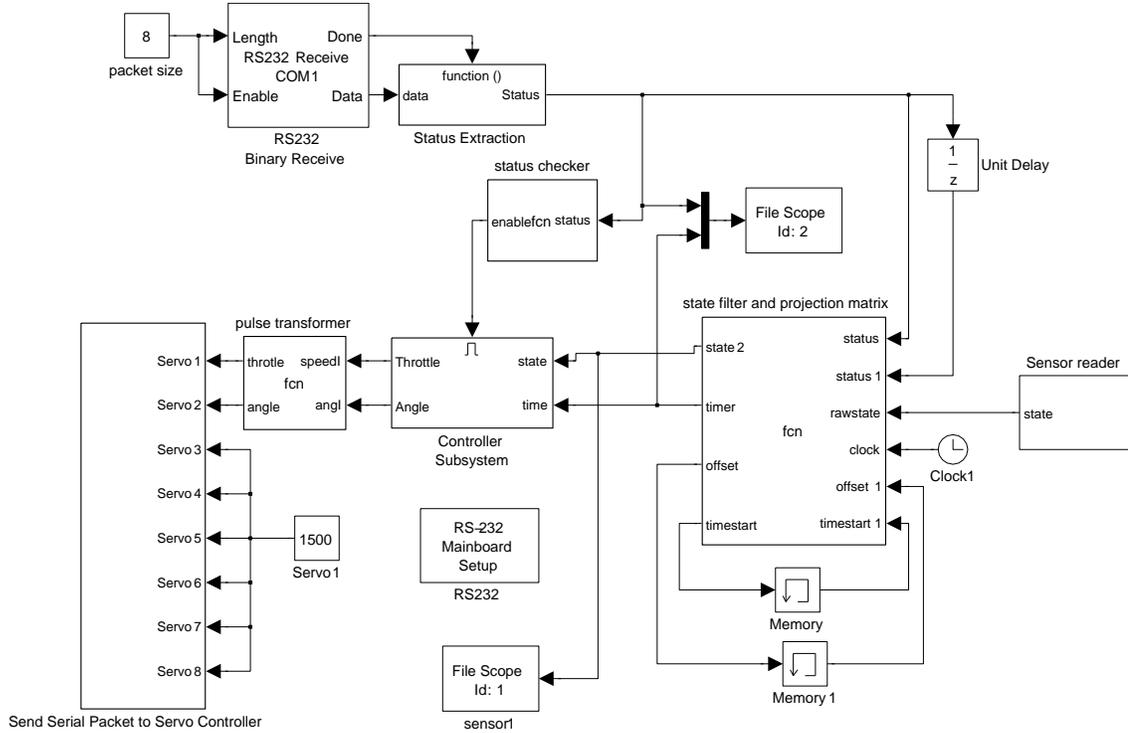


Figure 5.1: Implementation Interface

5.2 Differential Coordinate System

Within the controller subsystem a component is added to smooth the trajectory planning of the controller by creating a smoothed third order polynomial path between the outputs at start, or trigger, time and the final desired system outputs (l_{12}, ψ_{12}). The smoothing polynomial is designed to work for the first 8 seconds upon triggering. The designed polynomial term D is used to replace the desired values of the system outputs, i.e. $\mathbf{D} = [l_{12}^d; \psi_{12}^d]$, and similarly for the derivative terms. Recall that previously the outputs were defined as \mathbf{z} in Eq. 2.22. Here a similar notation is used for

the desired and actual outputs, that is, the initial desired output $\mathbf{z}_d = [l_{12}^d; \psi_{12}^d]$. This and the initial actual output (\mathbf{z}_o) are used to define the polynomial as follows:

$$\begin{aligned} \mathbf{D} &= \mathbf{a}_5 t^5 + \mathbf{a}_4 t^4 + \mathbf{a}_3 t^3 + \mathbf{a}_0 \\ \dot{\mathbf{D}} &= 5\mathbf{a}_5 t^4 + 4\mathbf{a}_4 t^3 + 3\mathbf{a}_3 t^2 \\ \ddot{\mathbf{D}} &= 20\mathbf{a}_5 t^3 + 12\mathbf{a}_4 t^2 + 6\mathbf{a}_3 t \end{aligned} \tag{5.1}$$

where terms \mathbf{a}_i are vectors defined from the initial values by:

$$\begin{aligned} \mathbf{a}_5 &= 3(\mathbf{z}_d - \mathbf{z}_o)/16383 \\ \mathbf{a}_4 &= -20\mathbf{a}_5 \\ \mathbf{a}_3 &= 320\mathbf{a}_5/3 \\ \mathbf{a}_0 &= \mathbf{z}_o \end{aligned}$$

and as mentioned, the terms \mathbf{z}_d and \mathbf{z}_o are the final desired and actual initial system outputs respectively.

This smoothing function combined with the resetting of the position of the boat upon triggering removes most of the jerk, or impulses, that occur in practice when triggering the controller, by reducing the tracking error effectively to zero when the system is triggered.

5.3 Experimental Setup

Prior to running the controller and boat in closed loop testing, the boat parameters are determined through grey box identification. The boat is run in open loop while

recording the boat state outputs from the sensor. These outputs are then compared to the simulated outputs in an iterative approach and through reducing the error between the actual and simulated boat outputs the actual boat parameters are estimated.

Several open loop runs are done to collect data sets for this identification procedure. The first two cases are with a ramped up, held, then ramped down propeller speed and constant rudder angles of $\pi/6$ and $\pi/8$ radians. The speeds for both linearly ramp up, and down, to a speed of 130 rotations per second (rps), over a 10 second period at the beginning and end of the run, holding the top speed for 30 seconds. The third open loop run held the angle constant at zero and steps the speed between four values for 15 seconds each. The four speeds are 50 rps, 130, 90, and 0 rps in that order. The fourth case is with constant speed of 100 rps and an alternating rudder angle switching between zero, positive $\pi/6$, zero, negative $\pi/6$ and finally back to zero again, holding the angle for 15 second periods as with the third test. The inputs are shown in Fig. 5.2

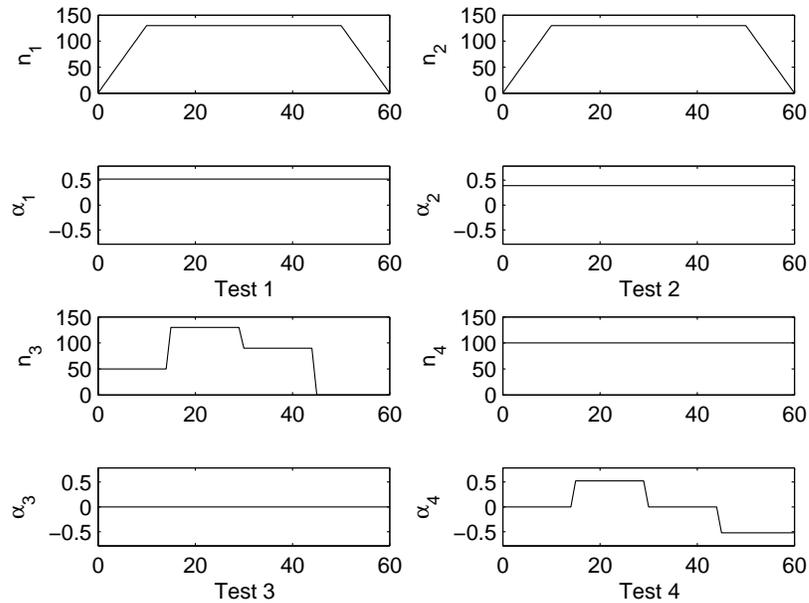


Figure 5.2: Inputs for Identification Tests

The initial set of open loop testing was done at Rundle park. The tests were each run after a power cycling of the computer and sensor and waiting 3 minutes for the sensor to acquire a good satellite count for GPS, to prevent any drift in the position values. Later batches of tests were done at Hawrelak park and Okanagan lake after further modifications to the boat were made.

The following figures, Fig. 5.3 and Fig. 5.4 show example results obtained from the open loop runs of test 1. The first depicts the usual trajectory of the boat in a circular case, while the second depicts the velocities recorded corresponding to this sample. In these tests, the orientation read from the sensor is not reliable. Therefore the u and v readings from the sensor are assumed to be unreliable as well. The orientation is corrected offline and the u and v are calculated based on the GPS outputs and the corrected orientation. The corrected results are compared to simulated results shown in Fig. 5.4 by the solid line.

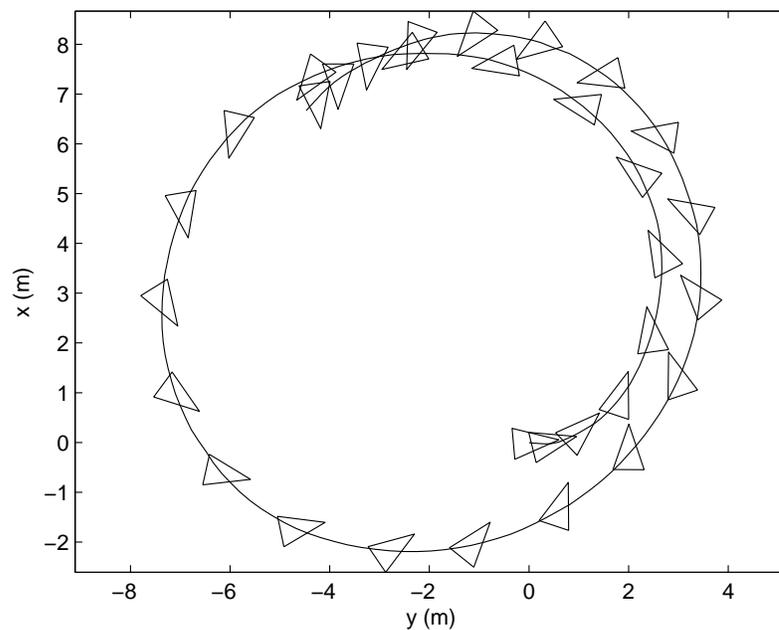


Figure 5.3: Open Loop Boat Trajectory from Test 2

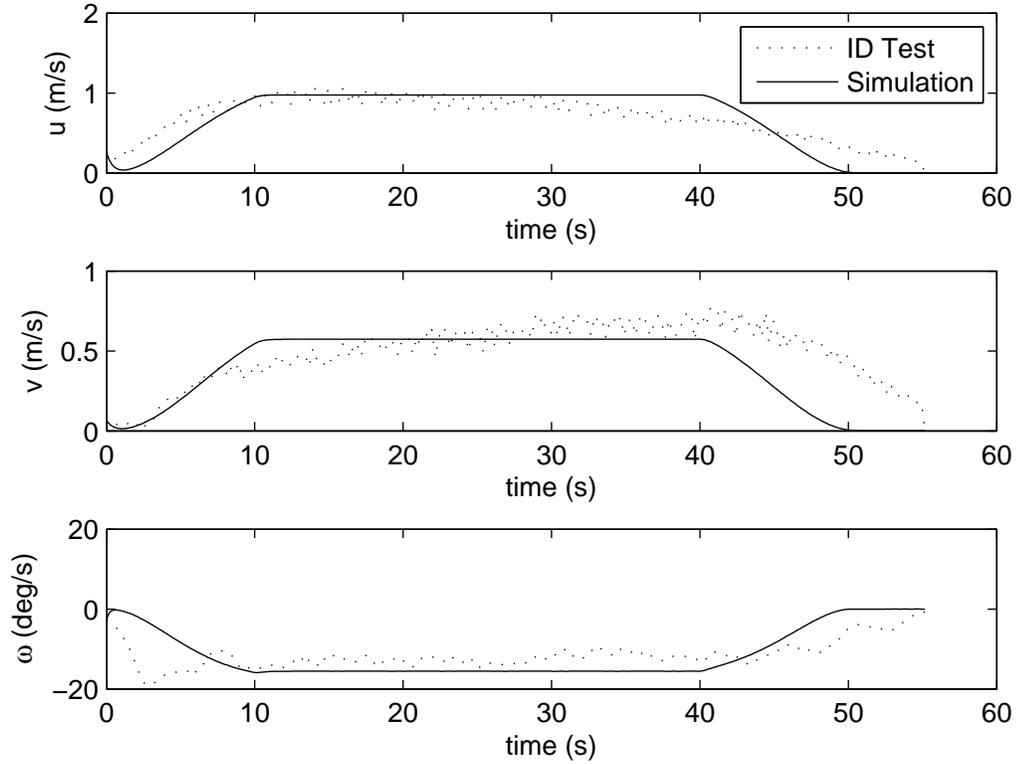


Figure 5.4: Recorded and Simulated Velocities for Test 2

After obtaining the open loop data sets the identification algorithm is used to determine the boat parameters m_{ij} , d_{ij} . The algorithm used is based on grey box identification. In grey box identification the parameters of a physical model with a known mathematical form are obtained by minimizing the least square difference of states of the simulated model and the test results. For our case, the form of the 3D dynamic model of the boat is the same as introduces in [5] and shown in Chapter 2. The boats dynamic equations (Eq. 3.1, and following loading relations) relating the system outputs and parameters to the inputs are redefined in terms of three error terms. One term for each of the dynamic motion equations as described in equation

5.3 is considered to define the least square error as:

$$\begin{aligned}
e = & \frac{1}{2} \sum_{i=1}^n (m_{11}\dot{u}_i - m_{22}v_i r_i + d_{11}u_i - \rho D^4 K_t U_{1i})^2 \\
& + \frac{1}{2} \sum_{i=1}^n (m_{22}\dot{v}_i - m_{11}u_i r_i + d_{22}v_i - \rho D^4 K_t U_{2i})^2 \\
& + \frac{1}{2} \sum_{i=1}^n (m_{33}\dot{r}_i - (m_{11} - m_{22})v_i u_i + d_{33}r_i - \rho D^4 L K_t U_{2i})^2
\end{aligned} \tag{5.2}$$

where U_1 and U_2 are the input components corresponding to $n|n| \cos \alpha$ and $n|n| \sin \alpha$, respectively.

The partial derivative of this error with respect to the unknown model parameters m_{22} , m_{33} , d_{11} , d_{22} , d_{33} and k_T are put to zero. The derivatives provide seven equations for the seven unknowns. Note that the primary mass term m_{11} is not included with the parameters listed. The reason for this omission is that all the parameters are scaled proportionately by this value, thus the values deduced through identification are dependent on the choice of m_{11} . Scaling m_{11} will proportionately scale the other parameters and does not affect the behaviour of the dynamic system. The seven partial derivative equations are then rewritten in the form $\mathbf{A}\vec{\mathbf{x}} = \vec{\mathbf{B}}$. Then the system responses, samples \dot{u}_i , \dot{v}_i , \dot{r}_i , u_i , v_i , r_i and inputs U_{1i} and U_{2i} are used to assemble the values of the \mathbf{A} and $\vec{\mathbf{B}}$ matrices. Finally, the nominal parameters can be determined by:

$$\vec{\mathbf{x}} = \mathbf{A}^{-1}\vec{\mathbf{B}} \tag{5.3}$$

From the identification the results found (based on the listed value for m_{11}), that are subsequently used in the closed loop testing of the controller, are:

$$m_{11} = 7.80 \quad d_{11} = 20.679$$

$$m_{22} = 8.394 \quad d_{22} = 20.492$$

$$m_{33} = 0.414 \quad d_{33} = 2.197$$

$$K_t = 0.133$$

With these parameters, simulation of the boat model produces output values that are compared to the recorded values of the actual system. Refer to Fig. 5.4, the simulated state, depicted by the the solid line, matches the recorded data, shown by the dotted line. These parameters were also simulated with the other test data taken and are shown in Figures 5.5, 5.6, and 5.7.

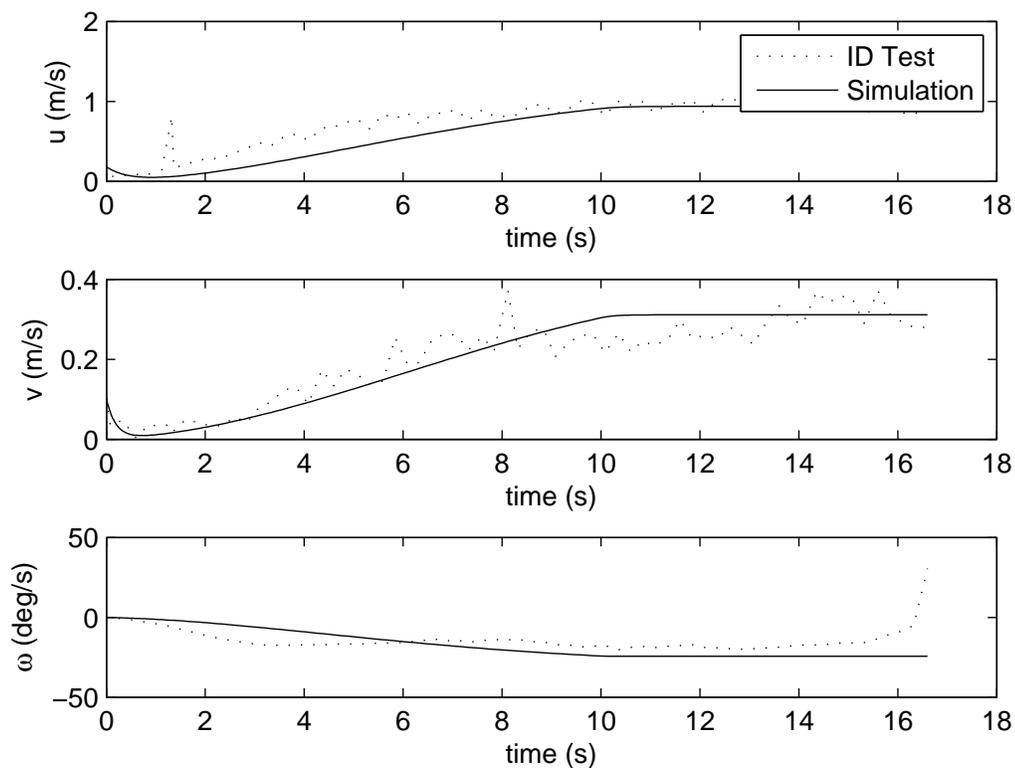


Figure 5.5: Recorded and Simulated Velocities for Test 1

5.4 Closed Loop Setup

With the parameters determined and entered into the controller algorithm, the system becomes ready for closed loop testing. Since the model parameters may be velocity dependant, the desired trajectories of the leader are defined such that the follower operates at the velocity range used in the open loop identification tests. Specifically

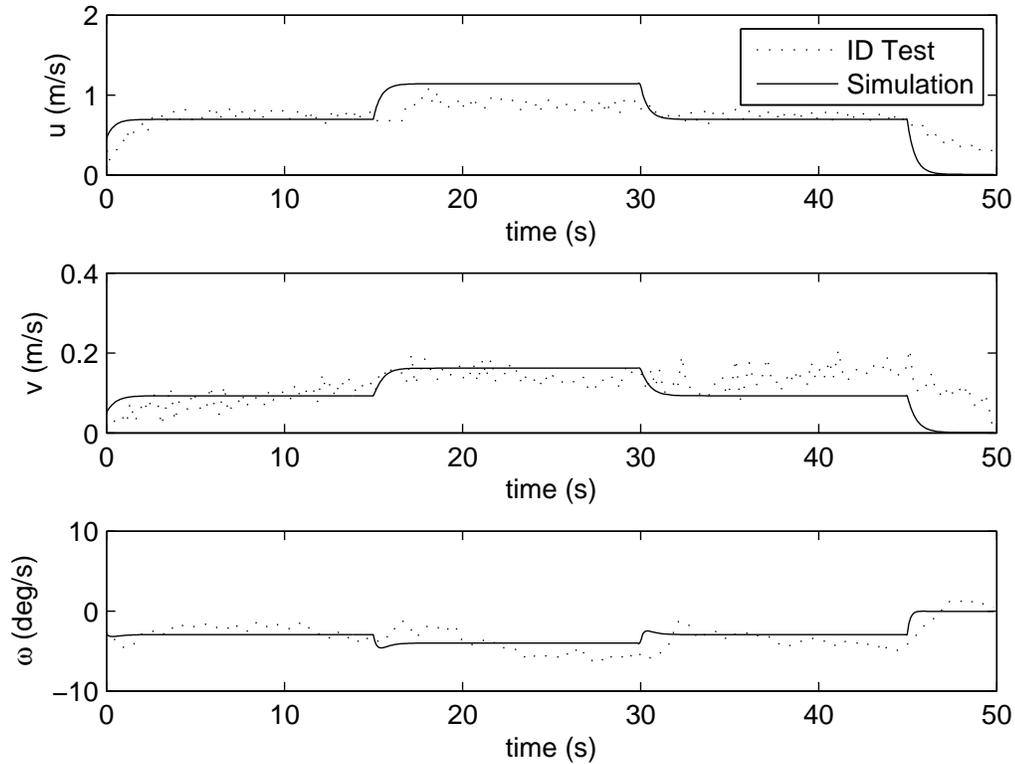


Figure 5.6: Recorded and Simulated Velocities for Test 3

there is a straight lead path case and a circular lead path case. The desired formation the controller sets is to locate the follower 90 degrees to the right of the leader. Thus causing the boats, for most cases, to have parallel trajectories. Prior to the water tests, the system is dry-run to check the outputs responses and triggering of the program. Additionally, the batteries are checked, and charged fully if at a low Voltage.

For the water tests a tether is used as a fail safe in case of power loss or loss of control with transmitter. The system is activated and program is started as with the open loop testing procedure. Similarly, the boat is driven manually out onto the water a safe distance away from any obstacles and the shore. The boat is positioned in such a way as to try and match the initial orientation to that of the desired one for

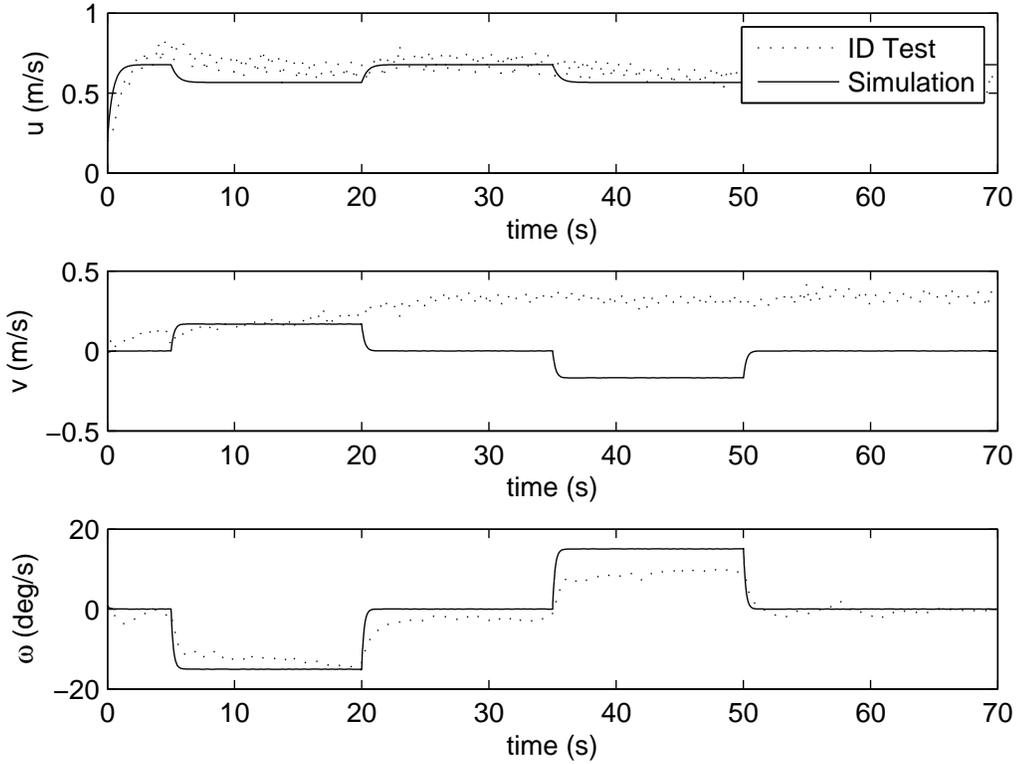


Figure 5.7: Recorded and Simulated Velocities for Test 4

following the virtual leader boats. For most cases this meant starting with a north heading. The velocity is also manually set to be as close to zero as possible, with any movement in the forward (aligned with the desired trajectory) direction.

Some assumptions made for these closed-loop tests are that there is no disturbances due to wind, waves, or current. The controller is tuned to handle a steady disturbance of up to 3 m/s through the uncertainty of the parameters (refer the controller gain derivation in chapter 2), but is not explicitly designed to handle a steady disturbance in any specific direction. For all the closed loop water tests there is a slight to moderate disturbance from either wind, current or wave sources.

5.5 Experiment Results and Discussion

The first set of closed loop tests are done at Harwreлак park. Of the closed loop tests done at this location, only the straight case has a reasonable result. The best response is shown in Fig. 5.8 and occurred on a day with above average wind speed. The pond location in this park is shielded from most wind directions, however at that time the wind strength and direction was enough to cause a noticeable drift on the vessel. Also shown are the controller outputs and velocities of the test in Fig. 5.9 and Fig. 5.10.

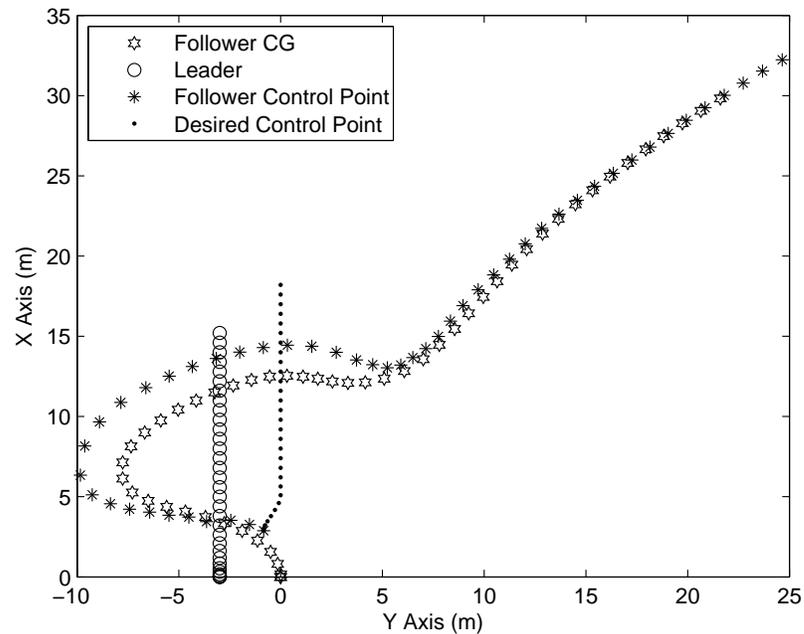


Figure 5.8: Closed Loop Boat Response: Trajectory

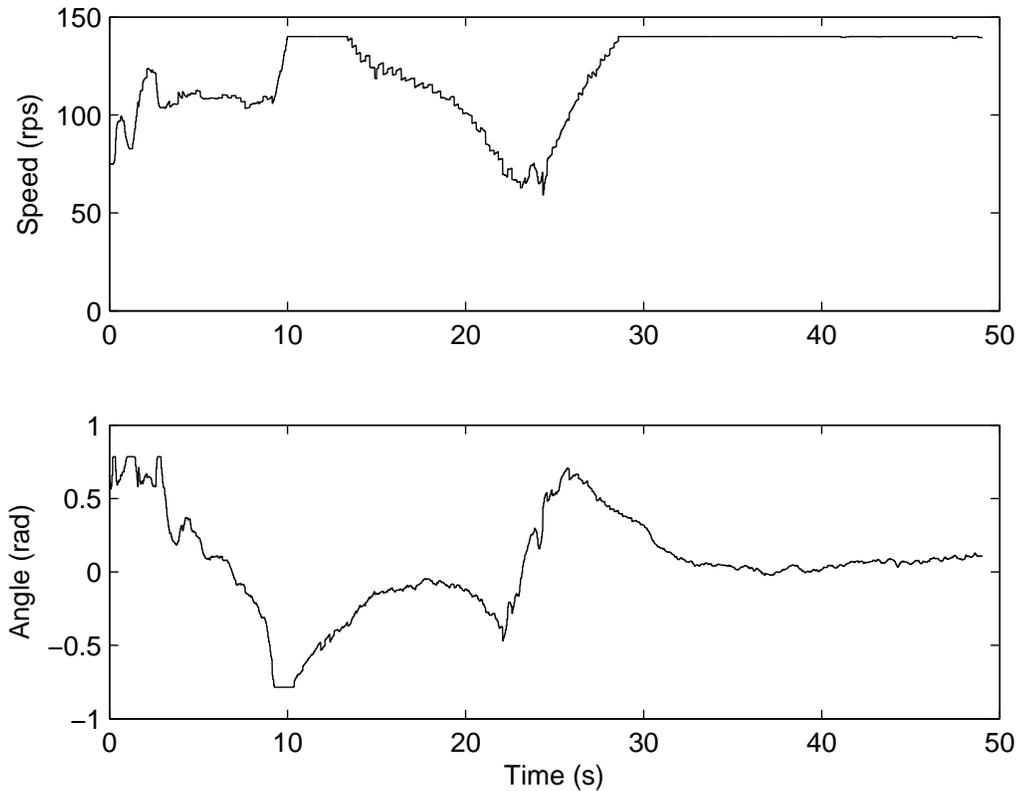


Figure 5.9: Closed Loop Controller Outputs; with saturation levels of 140 rpm and ± 45 degrees

Although this result does not follow the desired leader trajectory very well, it does exhibit the same behaviour as a simulated system given the same initial conditions and the same controller outputs. The simulated trajectory shown in Fig. 5.11 is produced using the same controller settings as the closed-loop test. That is, the system is simulated in open loop, but using the inputs recorded from the closed loop test. The controller settings are: $\lambda_1 = 0.1$, $\lambda_2 = 0.1$, $\eta = [0.1; 0.1]$, saturation buffer $\phi = [0.1; 0.1]$, and control point distance $d = 3$ meters. This simulated trajectory, and the velocities, Fig. 5.12, are indicating that the model for the boat is roughly accurate but that better control of the initial position and orientation are needed to produce the desired formation/trajectory. The reason for this is that the error in the

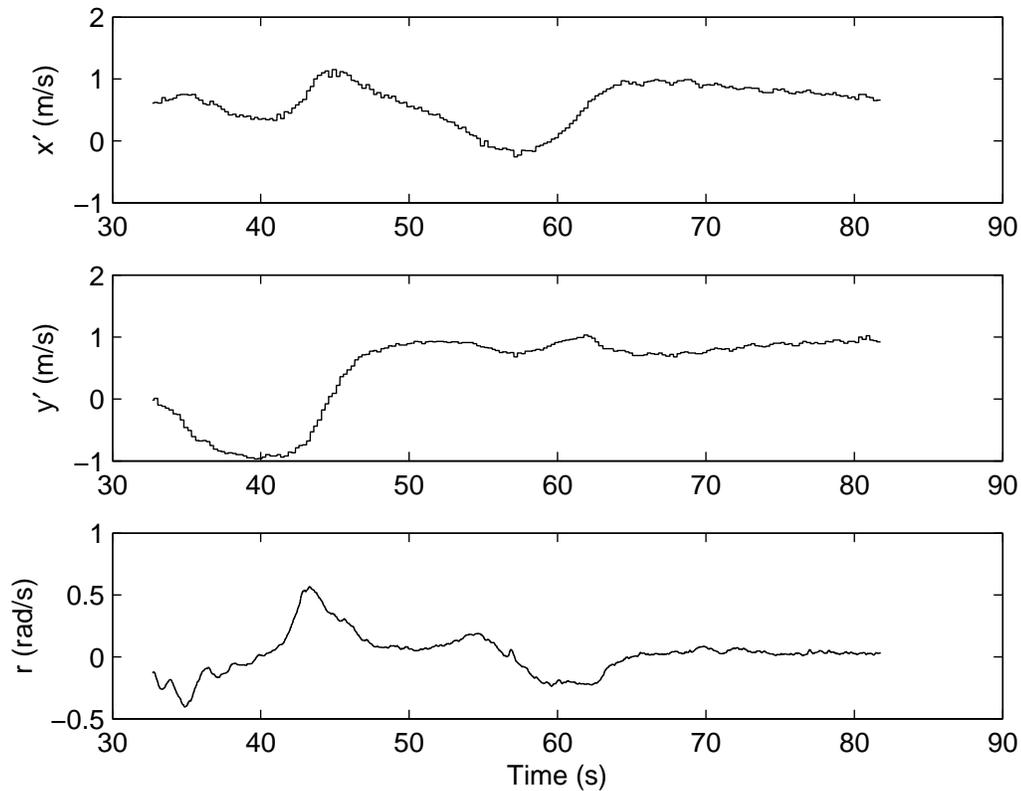


Figure 5.10: Closed Loop Boat Response: Velocities

system outputs (l, ψ) occurring at the time of triggering the controller is such that the controller outputs saturated the available responses of the system. The saturation limits imposed are 140 rps for the speed and 45 degrees for the angle (positive or negative). Refer to Fig. 5.9, specifically the angle outputs at time of zero. The boat has physical limitations to propeller speed and rudder angle. The simulation in Fig. 5.11 use the same initial conditions and saturation limits imposed by the physical vessel. The continuation of the saturation of both the speed and angle, well after the triggering, also indicate that the controller has not yet been tuned properly. Further, the simulated responses do not include any disturbances, if the actual disturbances could be accurately included into the simulated model, the result may have been even closer to the actual. Nor does the simulated model include the added damping and

torque introduced by the tether dragging in the water behind the vehicle.

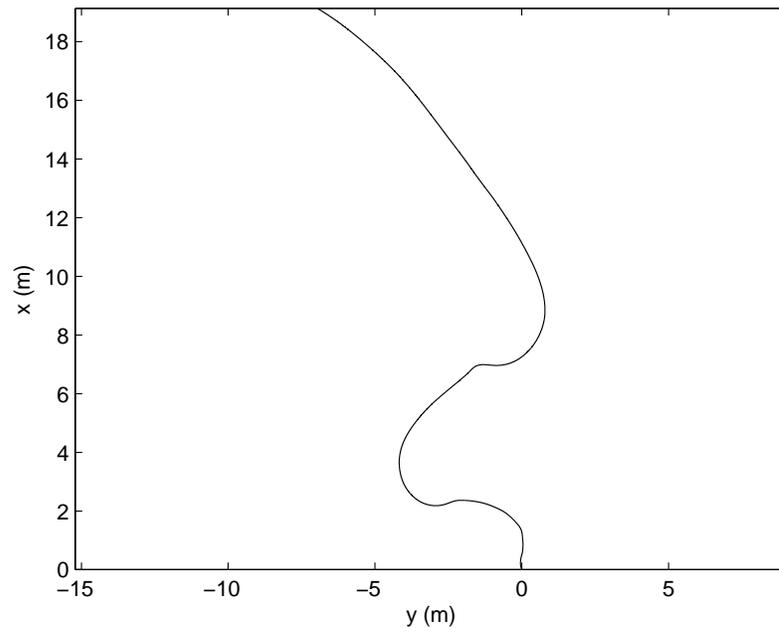


Figure 5.11: Simulation of Boat Response: Trajectory

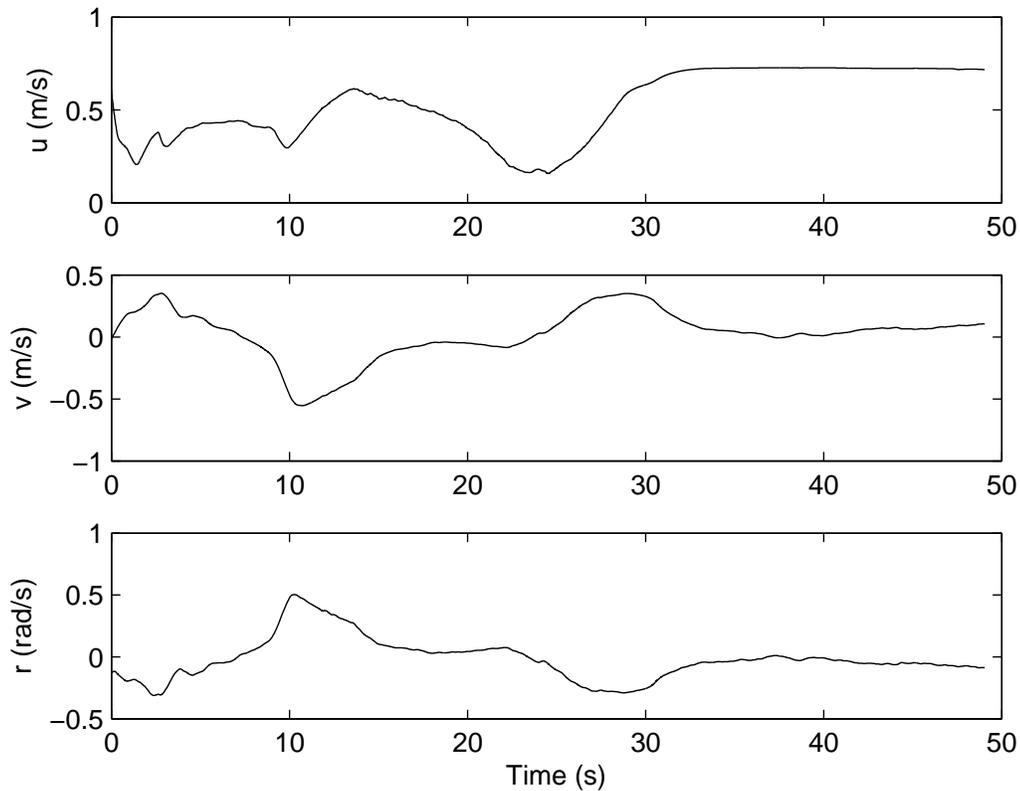


Figure 5.12: Simulation of Boat Response: Velocities

From this result, based on poor initialization of the system at triggering, a PID component is added to the controller. The PID component runs for a set time, upon triggering of the system, to get the boats orientation and velocity to the desired values. Then the formation controller would start with, ideally, zero error for the controller inputs. The closed loop testing incorporating this PID component took place on Okanagan lake. This is a much larger body of water and thus is less protected from disturbances such as wind and waves.

Figures 5.13 and 5.14 show example results obtained for the Okanagan tests and as can be seen the closed loop control of the boat is unstable and does not exhibit any of the desired behavior when using the sliding mode formation control scheme. Note that the PID section of control, lasting for the first thirty seconds of recorded values,

and indicated by the asterisk in the trajectory figure does not counter any disturbance. It only is controlled such that the speed in the forward direction and orientation angle reaches the desired values for initialization/switching to sliding mode control. Thus the diagonal trajectory, and the nonzero v velocity are indications of the magnitude of disturbances during this test.

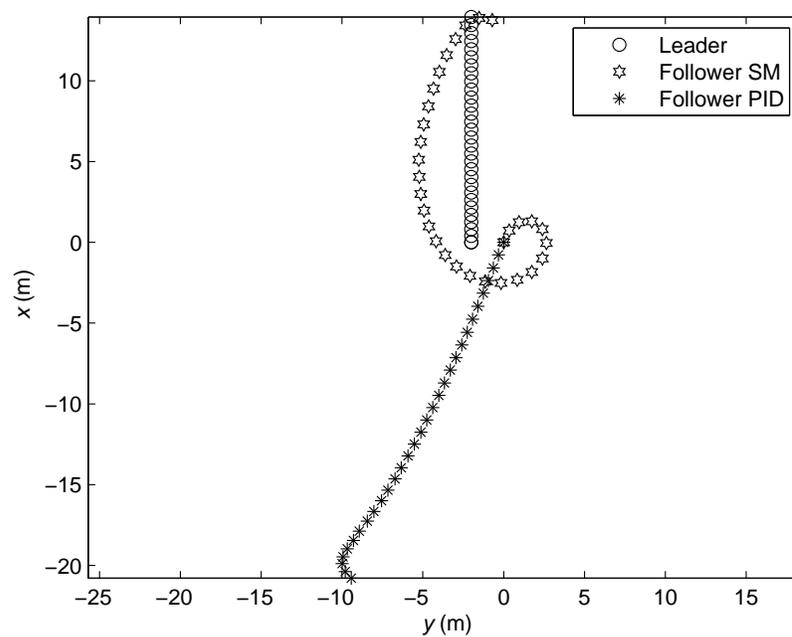


Figure 5.13: Closed Loop Boat Response: Trajectory

The problems that are the most likely culprit regarding this instability and uncontrolled behaviour of the vessel are covered further in the next chapter along with work done to overcome them.

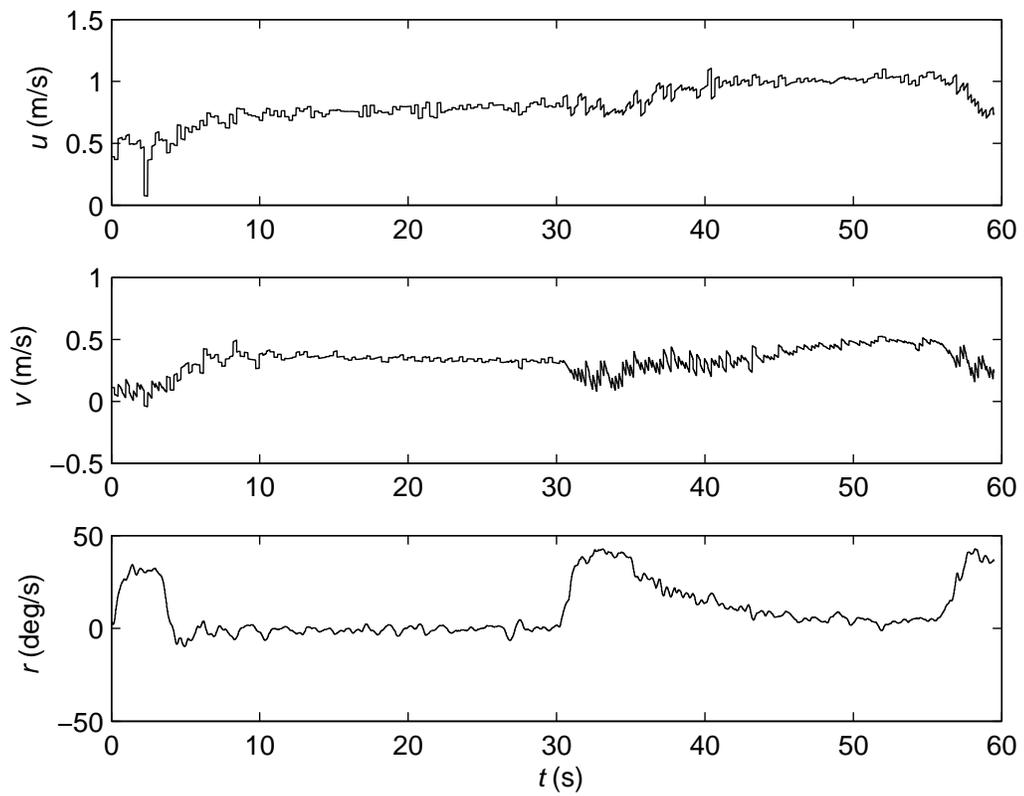


Figure 5.14: Closed Loop Boat Response: Velocities

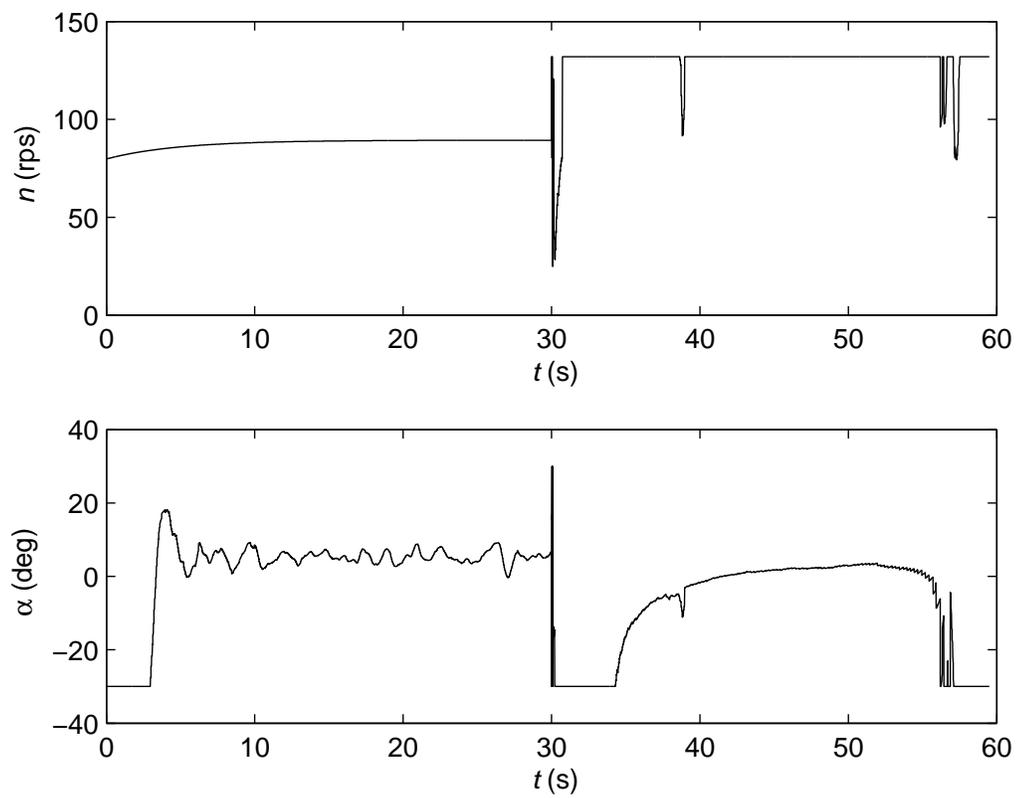


Figure 5.15: Closed Loop Boat Response: Controller Inputs

CHAPTER 6

EXPERIMENTAL RESULTS - ENHANCED MODEL

As mentioned in the last chapter there are several possible culprits behind the instability and uncontrolled behaviour of the boat in the closed loop tests. The largest perceived problem is due to differences between the physical boat and the vessel model. To rectify this, the boat model is updated along with the propeller thrust equation. Complementary to this, the grey box identification is done again for the new model, with a more robust program, to better tune the parameters used. These steps are covered in the following sections as well as some simulation results comparing the new model to the old one and to the actual data obtained from open loop identification tests. The other possible problems are inaccuracies of the sensor and the combined uncertainty from disturbances, from the identification process, and while running controller, being greater than the uncertainty the controller is designed to handle.

6.1 Update of Boat Model

The first amendment is to update the boat model to account for a more complex boat hull. The prior boat model had assumed fore-aft symmetry as well as a top-bottom symmetry where as the actual boat has neither. Furthermore the bilateral symmetry is also not exactly the same as the pontoons have a slight warp to the wood and

thus are not identical. However this symmetry is still assumed, as the differences are significantly less than the others. To account for these non symmetrical aspects of the vessel hull coupling terms are incorporated into the dynamic equations of the vessel. Eq. 6.1 is the updated equations of motion including the coupling terms m_{23} , m_{32} , d_{23} , and d_{32} .

$$\begin{aligned}
 m_{11}\dot{u} - m_{22}vr - \left(\frac{m_{23}+m_{32}}{2}\right)r^2 + d_{11}u &= X \\
 m_{22}\dot{v} + m_{11}ur + m_{23}\dot{r} + d_{22}v + d_{23}r &= Y \\
 m_{33}\dot{r} + (m_{22} - m_{11})uv + \left(\frac{m_{23}+m_{32}}{2}\right)ur + m_{32}\dot{v} + d_{32}v + d_{33}r &= N
 \end{aligned} \tag{6.1}$$

It should be noted that with the configuration of the boat using pontoons it is possible to have negative values for the mass coupling terms m_{23} and m_{32} but not for the new damping terms [5].

6.2 Update of Propeller Thrust Relation

The second amendment is the update of the propeller thrust relation. For the initial thrust equations the propeller thrust coefficient K_T was assumed to be constant. However this simplified case does not accurately portray the actual case. For most applications the propeller thrust coefficient is a non-linear term [5] given by:

$$K_T = \alpha_1 + \alpha_2 J_0 \tag{6.2}$$

where α 's are constant parameters and J_0 is the advance coefficient. This coefficient is given as:

$$J_0 = \frac{V_a}{nD} \tag{6.3}$$

where V_a is the advance speed. This speed is related to the boat speed by:

$$V_a = (1 - \omega)u \quad (6.4)$$

here ω is the wake factor and is a constant between 0.1 to 0.4.

These equations are combined with the initial thrust equation. Then simplified by combining unknown terms α_2 and $(1 - \omega)$ into a single α_2 value. Thus, the updated thrust equation resulting is:

$$T = \alpha_1 U_1 + \alpha_2 U_2 \quad (6.5)$$

Here $U_1 = \rho D^4 n_2$ and $U_2 = \rho D^3 u |n|$, unless $n = 0$ in which case $U_2 = 0$ to deal with the singularity in the advance coefficient term (Eq. 6.3).

A further consideration for this update is the non-linear relationship between the propeller speed n and the pulse width. Considered, is the replacement of the relation with an unknown parameter set as follows:

$$n = -Ap_w^2 - Bp_w + C \quad (6.6)$$

where p_w is the pulse width and the unknown parameters are A, B, C . However it is found that this created too many unknowns for the identification procedure, causing it to become unsolvable with the grey-box identification method chosen. Thus this amendment is not used.

6.3 Parameter Identification with Updated Model

For the retake of data for identification purposes, done at Okanagan lake, the identification open loop tests were simplified to a single case. A ramping up, then steady, then ramped down speed, with a constant non-zero angle. This differed from the previous suite of identification tests by the omittance of the straight and zigzag shaped

trajectory cases. Earlier figures (Figs. 5.3 and 5.4) show similar prior open loop examples, of the trajectory and velocities recorded, and later figures (Figs. 6.2 through 6.4) depict velocities for the Okanagan identification tests.

The new parameter identification procedure further differed from the previous method of grey- box identification. Previously a M-file was used, that performed reverse kinematics, iterations, etc. that make up grey box identification, to determine the parameters. For the new identification process MATLAB's grey-box identification toolbox was utilized.

The use of this toolbox required an initial assumption, or value(s) for the parameters and a range for each parameter. Additionally, it is possible to set any of the parameters to be fixed to the initial value (listed in Table 6.2). With these constraints, the program perturbs the parameters and determines if the perturbed values are better or worse than the originals. This method is then iterated until some minimum cost, or trigger, at which the optimum parameter values are obtained. This toolbox was run with five differing constraint sets, as changing the programs constraints produced unique results. Refer to Table 6.3 showing an example of the results of this program. The five constraint sets are due to a combination of set and range constraints. These different constraints are listed in Table 6.1 and Table 6.2.

These results are then compared by running the parameters in a simulated boat model. The error between the simulated velocities and the recorded open loop velocities is summed along with the errors in orientation to give each parameter set an error value. Ie. $\int_{t=t_i}^{t_f} [|\tilde{u}(t)| + |\tilde{v}(t)| + |\tilde{r}(t)| + |\tilde{\theta}(t)|] dt$. These errors are compared for each parameter set across multiple open loop cases to determine the parameter set yielding the minimal error value. This parameter set is taken as the optimal one, best fitting the actual boat parameters. Table 6.4 shows the error values computed when

Table 6.1: Grey-box Identification Parameter Constraints

Case	S_1			S_2			S_3		
Parameter	Fixed	Min	Max	Fixed	Min	Max	Fixed	Min	Max
m_{22}	0	m_{11}	$5m_{11}$	0	m_{11}	$5m_{11}$	0	m_{11}	$5m_{11}$
m_{23}	1	0	50	0	0	50	0	-30	50
m_{32}	1	0	50	0	0	50	0	-30	50
m_{33}	0	0	10	0	0	10	0	0	10
d_{11}	0	0	30	0	0	30	0	0	30
d_{22}	0	0	50	0	0	80	0	0	50
d_{23}	0	0	50	0	0	50	0	0	50
d_{32}	0	0	50	0	0	50	0	0	50
d_{33}	0	0	10	0	0	10	0	0	10
a_1	0	0	1	0	0	1	0	0	1
a_2	0	$-\infty$	0	0	$-\infty$	0	0	$-\infty$	0

Table 6.2: Grey-box Identification Parameter Constraints Continued

Case	S_4			S_5			Initial Values
Parameter	Fixed	Min	Max	Fixed	Min	Max	
m_{22}	0	m_{11}	$5m_{11}$	0	m_{11}	$5m_{11}$	7.8
m_{23}	0	0	50	0	-30	50	0.0
m_{32}	0	-30	50	0	0	50	0.0
m_{33}	0	0	10	0	0	10	0.314
d_{11}	0	0	30	0	0	30	20.679
d_{22}	0	0	50	0	0	50	26.492
d_{23}	0	0	50	0	0	50	0.0
d_{32}	0	0	50	0	0	50	0.0
d_{33}	0	0	10	0	0	10	2.197
a_1	0	0	1	0	0	1	0.124
a_2	0	$-\infty$	0	0	$-\infty$	0	-1.0

comparing parameter sets to other data sets, as well as the data set with which the parameter set was derived from.

Note that for some cases the resulting error term is infinite. This is due to the simulated vessel becoming unstable, usually due to controller feedback attempting to reach infinite velocities, and thus the error terms are just assumed to be infinite as the simulation crashes without producing an actual error value. Also note that the Data Set # 43 did not produce parameter sets for all constraint collections. The reason for

Table 6.3: Parameter Results for Grey-box Identification with Data Set # 41

Constraint sets 1 - 5					
Parameter	S_1	S_2	S_3	S_4	S_5
m_{11}	7.8	7.8	7.8	7.8	7.8
m_{22}	8.855	7.907	18.271	7.8	22.497
m_{23}	0	0	0.451	0.016	-0.503
m_{32}	0	0.001	18.763	4.367	15.050
m_{33}	0.001	1.674	2.732	2.883	0.656
d_{11}	0.871	16.917	0.894	5.759	0.774
d_{22}	10.061	33.430	13.494	27.340	13.128
d_{23}	0.178	0.112	2.047	3.633	1.674
d_{32}	0.177	1.287	12.503	4.296	4.613
d_{33}	0.328	2.128	2.294	1.286	0.980
α_1	0.035	0.257	0.039	0.092	0.028
α_2	-0.19	-0.717	-0.419	-0.295	-0.195
Error vs. Set 41	3740.4	3738.7	3732	3733.5	3735.6

this was that the grey-box identification/optimization process was left to run for 24 hours and if it had not obtained a parameter set after that many iterations, that set was dropped from further consideration.

Additionally, the identification process is done a second time using the ramp down section of the open loop data. The original identification, as well as the identification just described used only the ramping up section at the beginning of the recorded data. As with the new procedure used with the ramp up section, the ramp down data is input into the grey-box program and parameter sets are obtained. These are put through the same error comparisons as well as compared against the ramp up errors and are found to be, for the majority, an order of magnitude greater than the ramping up errors, or else exhibiting undesirable (and unstable) trajectories, such as the infinite error values for the initial error calculations. Thus these parameter sets are not investigated further.

From these new identification open loop recordings, and the resulting grey-box identified parameters with correlating error values, the parameter set with the smallest

Table 6.4: Error Comparison across Data sets

Error vs. Data Set #	Parameter Set Derived From:				
	Set 41				
	S_1	S_2	S_3	S_4	S_5
41	3740.4	3738.7	3732	3633.5	3735.6
42	6761.1	4450.2	∞	73691	∞
43	11544	7967.7	∞	66243	∞
45	14191	4742	∞	1911.3	∞
Error vs. Data Set #	Set 42				
	S_1	S_2	S_3	S_4	S_5
41	3739.3	3731.7	3873.5	3735.8	10393
42	633.9	574.9	783.3	520.2	602.8
43	1882.1	55527	8654.7	3621.4	1914.6
45	2505.4	5897.1	32349	∞	3769.3
Error vs. Data Set #	Set 43				
	S_1	S_2	S_3	S_4	S_5
41	3739.2	3740	-	-	-
42	1406.2	2131.6	-	-	-
43	413.9	584.6	-	-	-
45	1836	1991.5	-	-	-
Error vs. Data Set #	Set 45				
	S_1	S_2	S_3	S_4	S_5
41	3736.9	3731.8	3735.3	3738.7	3731
42	1959.8	2611.4	2583.7	2970.2	2732.6
43	4080.2	1835.6	2090.3	2058.4	1892.6
45	43383	42369	42980	41184	42516

total error value, when compared to all the data sets used, is found to be the set produced by the first constraint configuration, and with data set 43. The parameters produced are the following:

$$\begin{aligned}
 m_{11} &= 7.80 & d_{11} &= 2.175 \\
 m_{22} &= 7.804 & d_{22} &= 11.837 \\
 m_{23} &= 0.0 & d_{23} &= 0.002 \\
 m_{32} &= 0.0 & d_{32} &= 0.692 \\
 m_{33} &= 0.346 & d_{33} &= 0.643 \\
 \alpha_1 &= 0.392 & \alpha_2 &= -5.009
 \end{aligned}$$

This set having an error of only 413.9 when compared to its father data set (#43), but also having the least total error when compared to other data sets #'s 41, 42, and 45 with errors of 3739, 1406, and 1836 respectively. Refer to Table 6.4.

6.4 Comparison of Models

Comparisons of the old versus new boat model are done through simulations of the models with the same inputs (n, α) as used for the open loop identification testing. The initial results comparing the three (the old, new, and actual) outputs are shown in Fig. 6.1, however there is still some difference between the model and actual system. Note that this can not actually be used to compare the models as this is the data set that was used to obtain the new boat parameters listed above. As such, it only shows the optimization of the parameters from the grey-box program and the effects, or lack thereof, of disturbances on the simulated outputs.

These disturbances are not explicitly taken into account during the identification process because they are assumed to be negligible, and they are very hard to determine or measure in real time. However, it is possible to simulate the system with approximated disturbances such as constant force for disturbance, and force applied

at CG of boat and thus not also introducing a disturbance torque to system. These assumptions are not realistic though. As an example of simulation with disturbance, the data set # 42 is used with a estimated disturbance of force 1 N at a heading of -45 degrees. The results generated are shown in Fig. 6.2.

For the three cases shown the loading terms n and α are similar to that of the identification, that is, they all have a ramp up, hold, then ramp down behavior for the speed while holding a constant angle. For the three cases shown (# 42, 43, 44) the angles are all $\pi/8$ radians, and the max speeds are 78, 96, and 111 rps respectively. Refer to Fig. 5.2 Test 2 for a input diagram similar to those used.

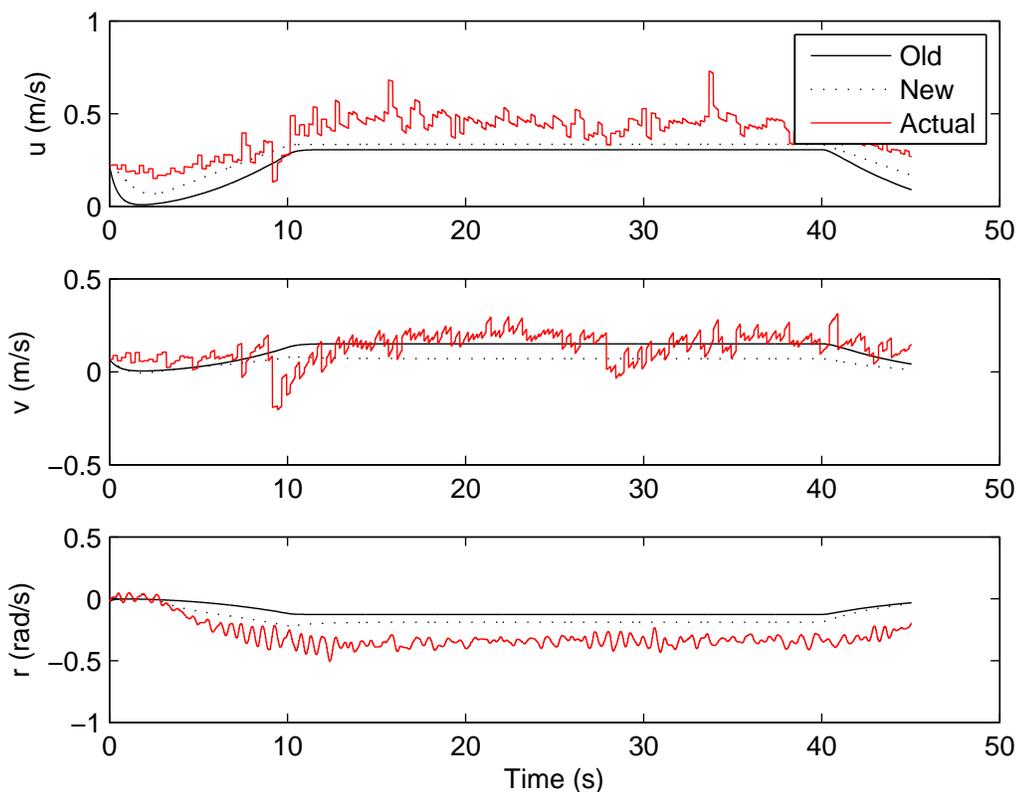


Figure 6.1: Comparison of Boat velocities for old model vs. new model vs. actual boat; with no simulated disturbances, from Data Set 43

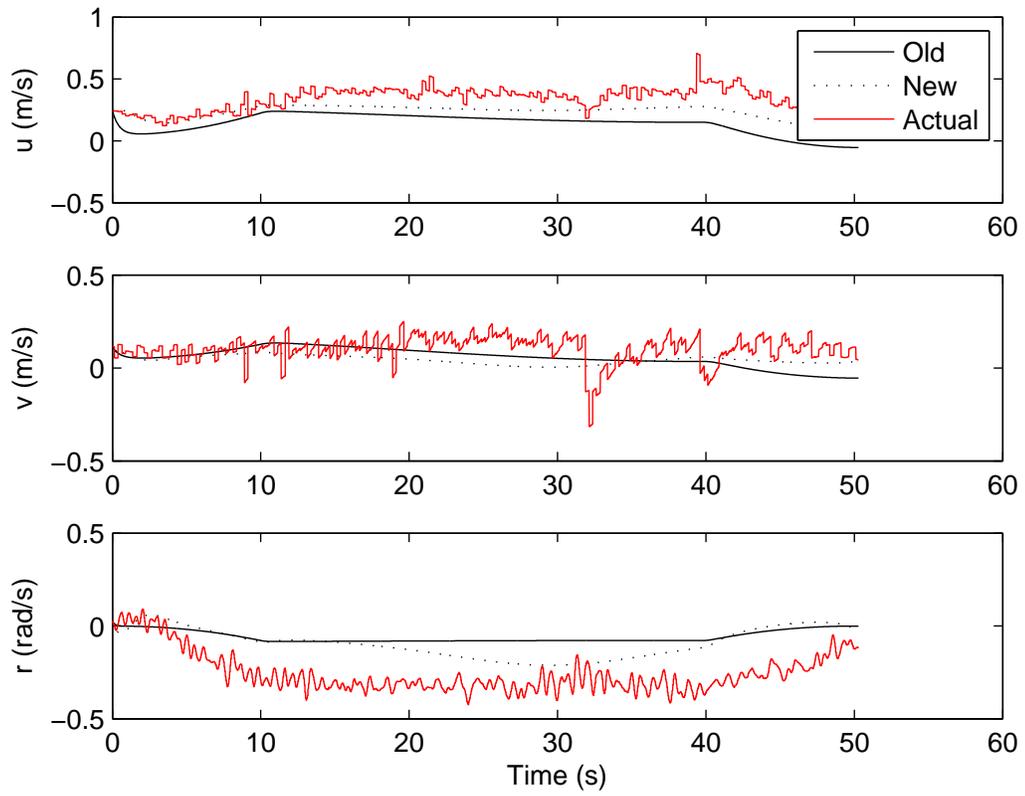


Figure 6.2: Comparison of Boat velocities for old model vs. new model vs. actual boat; with disturbances included, from Data Set 42

The following figures 6.3 and 6.4 show comparisons for local velocities of identification tests and the simulated models with added disturbance terms for data sets not used in the identification of parameters.

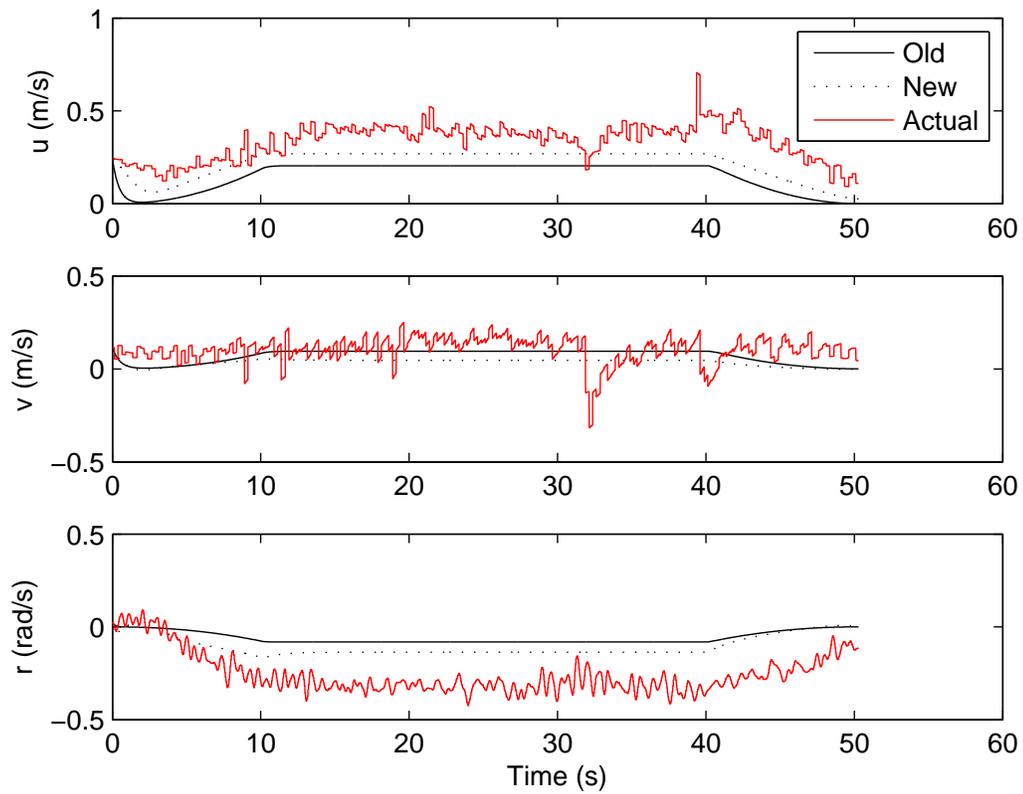


Figure 6.3: Comparison of Boat velocities for old model vs. new model vs. actual boat, Data Set 42

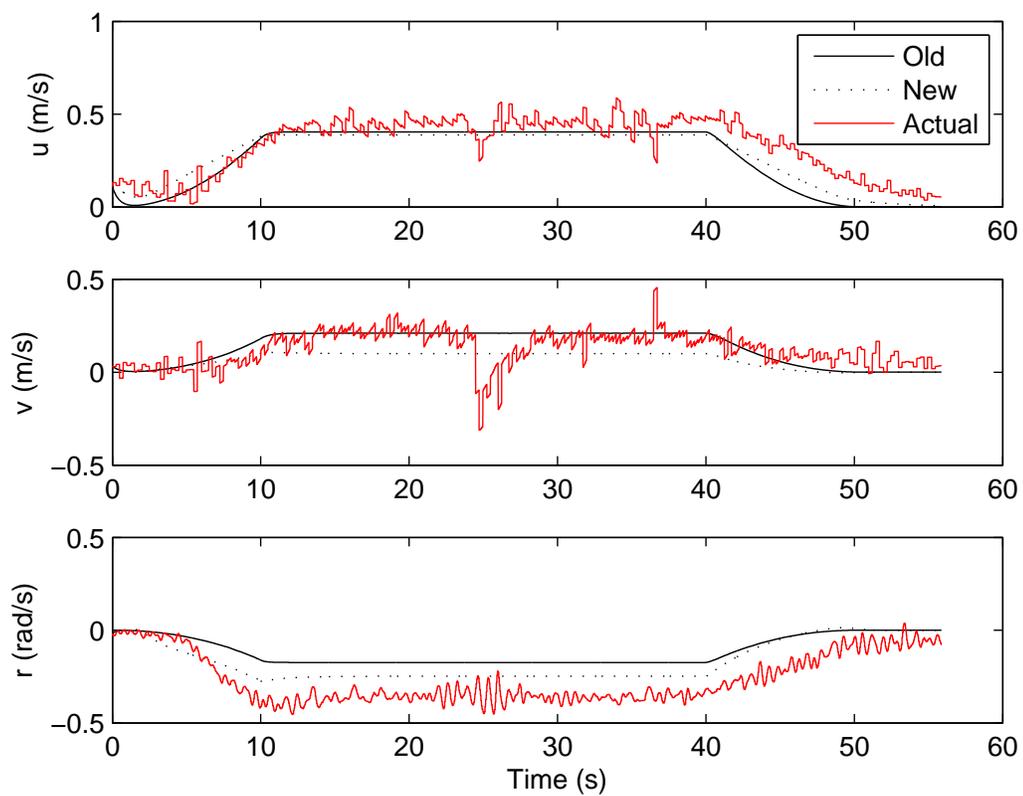


Figure 6.4: Comparison of Boat velocities for old model vs. new model vs. actual boat, Data Set 44

From the figures comparing the old and new simulated models we can see that the new, updated, model is better than the old one, as it closer fits to the recorded boat states.

6.5 Robustness of Enhanced Controller

In addition to the comparison of the old and new controller behaviors the robustness of the enhanced controller is determined. This is done through simulation of the updated system and varying the uncertainty of the parameters of the plant. Then the system outputs and boat responses are compared to the case with no parameter uncertainty. These errors are based on the 0% case having the true velocities. The speed-error values are calculated by:

$$\mathbf{SpeedError} = \sum |\dot{\hat{x}}| + |\dot{\hat{y}}| + |\dot{\hat{\theta}}| \quad (6.7)$$

for each time step. Note that this speed-error term does not correlate to any physical property (the units do not even match), this value is effectively the sum of the difference in areas, with absolute values, of the velocity response curves. Future work should be based on a normalized relation. Fig. 6.5 shows the speed error terms of the response. Similarly, the system outputs $l-\psi$ for varying uncertainties are compared for the same cases, are shown in Fig. 6.6. For both figures the 40% and 50% cases are highlighted with dashed lines.

From these figures it can be seen that the controller is robust. The system responses show little differences until the parameter uncertainty reaches 40%. The controller was designed to handle an uncertainty of up to 20% in the parameter values. That is, the controller is based on a 20% increase to the nominal values, thus leading to an expected theoretical robustness for uncertainties up to 20%. However, due to choosing conservatively when deriving the controller gain term ‘k’ values, refer

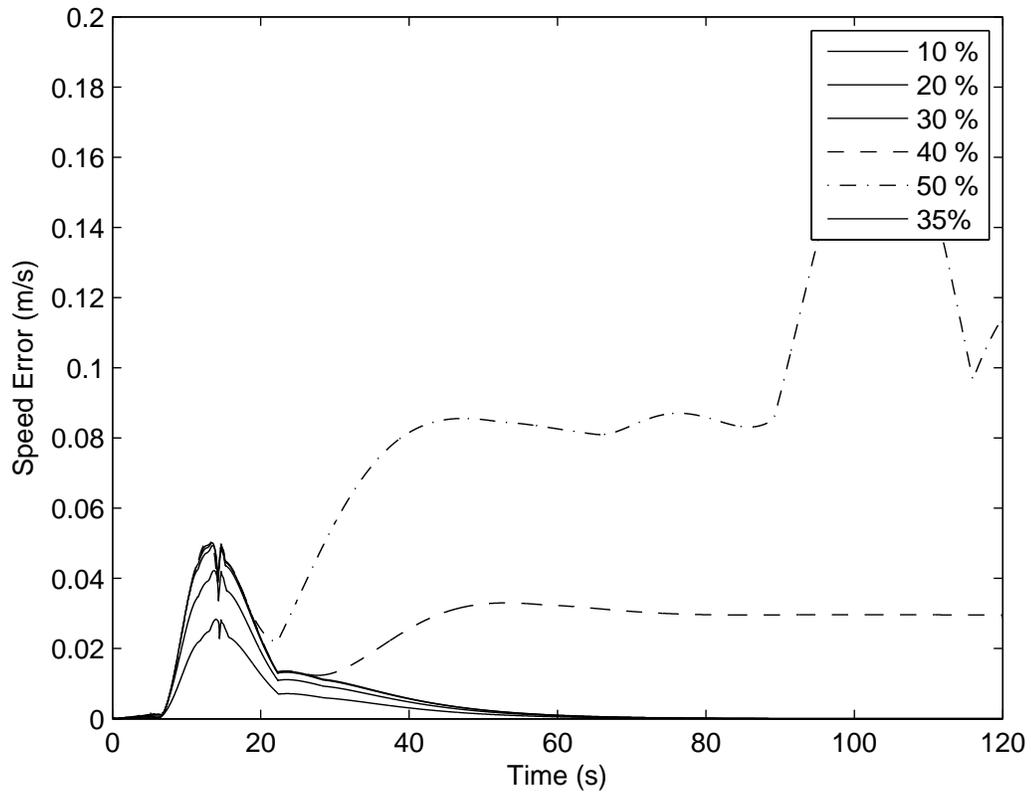


Figure 6.5: Sum of velocity errors for varying % uncertainty in parameter values

to Eq. 2.34, this resulted in extra robustness to uncertainties. Up to 40% for this particular simulation. Further, the small differences in l , or ψ for the cases less than 40% are due to using a saturation function instead of a sign function in the controller. This and the Saturation buffer size (here it is 0.01 for each surface) lead to these small differences. This proves the robustness of the updated control algorithm.

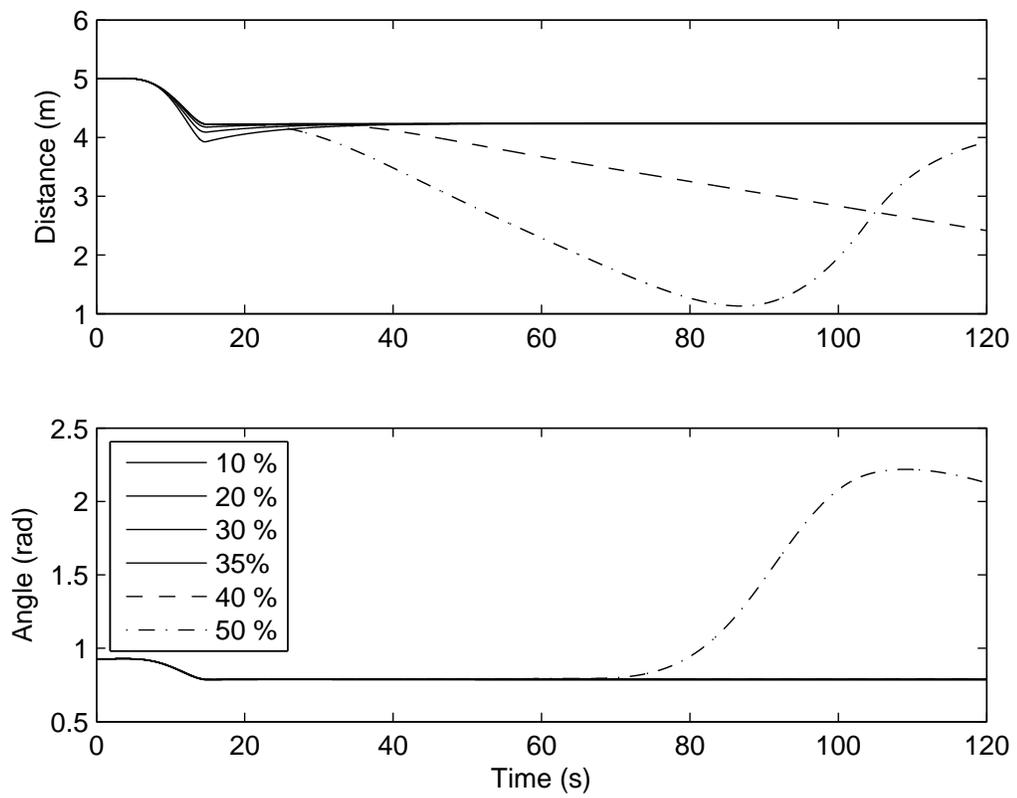


Figure 6.6: System outputs $l - \psi$ for varying % uncertainty in parameter values

CHAPTER 7

CONCLUSIONS AND FUTURE WORK

7.1 Contributions

- Developed formation controller using sliding mode control for a RC boat.
- Constructed development platform and interface for testing controller.
- Determined parameter values to match, as close as possible, the boat model to the actual RC boat dynamics.
- Showed robustness of controller system to large uncertainty of parameters (and disturbances).

7.2 Conclusions

A formation controller was developed, using a leader follower scheme, and sliding mode control upon the distance and angle between the CG of a lead boat and the control point of the follower vehicle. This controller was then implemented with a physical system using a PC/104 and sensor suite onboard a model RC boat. As the controller is based on the dynamic equations of motion of the vessel, the parameters of the simulation model of the boat needed to be determined. This was done with open loop testing and an identification procedure. Using these parameters to tune the

controller, closed loop testing was done with the boat and a virtual leader following a path, with a set formation.

From these tests it can be seen that the controlled system is not stable, nor exhibits the desired behavior while running under sliding mode control. These results can be seen in figures 5.13 and 5.14. The reasons for this failure of the controller can be accounted to the error, or mismatch, between the dynamic equations used (simulation model) and the actual vessels. Other sources of error were the simplified thrust assumptions, and errors in the parameter identification procedure.

Following these results, changes were made to the equations of motion to introduce coupling terms, as well as changes to the thrust equation. With these modifications the identification procedure was repeated with a new, more robust, method and new parameters were found. The parameter sets from this new identification method were examined and the set producing the least error was taken to be the set that most closely matched the actual vehicle. Simulations comparing the initial and updated model to each other and to the actual boat data (from open loop identification tests) were done. This is shown in figures 6.3 and 6.4. From these tests it can be concluded that the new updated model is an improvement over the initial one as it better matches the physical system response. That is, it produces less error when compared to the actual boat than the initial model.

As mentioned at the conclusion of chapter 3, the zero-dynamic stability of the vessel is also a problem. The root of the problem lies in the zero-dynamics of the vessel, namely that the vessel is underactuated and only two degrees of freedom are controlled. For this research the controlled DOF's are the position components (x, y) of the control point, while the orientation is indirectly controlled. By this the location (or control) of the control point affects the indirect control of the orientation [25].

Regarding the instability of the controller that occurs on occasions when the vessel is decelerating, the problem that happens is that the control point comes to a stop

but the CG of the boat does not. The result is the boat spinning around a point instead of coming to rest. A work-around for this problem was produced by setting the control point used by the controller to shift from ahead of the boat to the rear of the vessel depending on if the vessel is accelerating, or decelerating. This was implemented for simulations of the vessel as a way to remove undesired behaviour as the boat decelerated.

The reason that this works can be understood through a simple exercise of imagination. Imagine pulling the boat by the prow, no matter what the initial orientation, the boat will come to be aligned in a tangential direction to the direction of motion pulling on the prow. The natural damping and dynamics of the vessel cause the ship to align in the desired direction. Where as if one was to push the boat from the stern, the boat would only maintain the desired orientation if one was pushing through the CG. Further this system would be very unstable, the slightest disturbance would cause the boat to rotate around resulting in being pulled from the stern. Thus the setting of the control point can be considered as setting from where the boat is being pulled from, and thus which orientations are inherently stable.

This method of switching the control point location worked to remove the undesirable behaviour, but introduced some discontinuities to the controller when the control point instantly jumped six meters. Due to this discontinuity, this method was not utilized for the actual control of the vessel. Further research done regarding the stability of the zero-dynamics of the vehicle by Schoerling [25], covers the case for the $l - l$ control scheme. This could be reviewed and extended to develop a viable solution for the $l - \psi$ case.

An additional conclusion is that the enhanced controller is robust. This has been shown in the previous chapter. The uncertainty of the parameters of the boat were varied and it was shown that the controller had almost no sensitivity to parameter changes less than 40%. What little sensitivity to the parameters that was expressed

was due to the change from sign to saturation function in the controller and the choice of the size of the buffer region for the saturation function. This lack of sensitivity is equivalent to robustness and it should be noted that the % change in parameter values the controller can handle is almost double what it was designed for. This is due to conservative controller gain choices.

7.3 Future Work

The following are suggestions for future work to sort out, or circumvent, some of the difficulties and problems encountered in this research. First, use a load cell or some other force sensing device to measure the thrust of the boat motors. Use this to record the thrust to pulse width relationship. This should be done over a range of advance speeds to cover all applicable boat velocities. This relation can then replace the inaccurate thrust relationship, which is based on equations developed for large commercial vessels, not small models. As well, this should remove any uncertainty in the conversion between pulse width and rotation per second of the propeller.

Secondly, performing the identification procedure in a contained environment to get parameter results that do not incorporate any uncertainty due to disturbances. Thirdly re-tune and reprogram the PID controller section of the applicable model to be more flexible in the time used for PID to reach the desired parameters before switching to sliding mode. Additionally, this could be merged with another controller, or weighting function to control the heading angle as well as the velocity. As opposed to the current PID which only controls the velocity in a certain direction, and has no control over the heading of the vessel.

Lastly, the control scheme should be tested on multiple vessels. That is to say with a physical leader as well as multiple follower boats instead of the virtual leader as was done over the course of this research. Additionally, with multiple boats,

this control scheme could be included with the $l - l$ scheme as covered in [14] to demonstrate formation control of three or more vessels. As a part of this the state of the leader boats will have to be send/shared with the follower boats, this can be done by incorporating a second ethernet port and a wireless local area network (LAN) with which the boats can all broadcast their state (position, velocity, etc.).

BIBLIOGRAPHY

- [1] J. Desai, “A graph theoretic approach for modeling mobile robot team formations,” *Journal of Robotic Systems*, vol. 19, no. 11, pp. 511–525, 2002.
- [2] V. Kapila, A. G. Sparks, J. Buffington, and Q. Yan, “Spacecraft formation flying: Dynamics and control,” *Journal of Guidance, Control, and Dynamics*, vol. 23, pp. 561–564.
- [3] J. H. A.P. Aguiar, “Position tracking of underactuated vehicles,” in *Proceedings of the 2003 American Control Conference*, vol. 3, pp. 1988–1993, June 2003.
- [4] G. Roberts, “Trends in marine control systems,” *Annual Reviews in Control*, vol. 32, pp. 263–269, December 2008.
- [5] T. Fossen, *Guidance and Control of Ocean Vehicles*. Chichester, England: John Wiley and Sons Ltd., 1994. p168-171.
- [6] T. Vaneck, “Fuzzy guidance controller for an autonomous boat,” *IEEE Control Systems Magazine*, vol. 17, pp. 43–51, April 1997.
- [7] R. S. A. T. W. Naeem, T. Xu, “The design of a navigation and control system for an unmanned surface vehicle for environmental monitoring,” *Proceedings of the Institution of Mechanical Engineering Part M: Journal of Engineering for the Maritime Environment*, vol. 222, no. 2, pp. 67–79, 2008.

- [8] H. Ashrafiuon and P. Ren, "Sliding mode tracking control of underactuated surface vessels," *Proceedings of the ASME Dynamic Systems and Control Division 2005*, vol. 74, no. 1 part A, pp. 11–16, 2005.
- [9] F. Fahimi, "Sliding-mode formation control for underactuated surface vessels," *IEEE Transactions on Robotics*, vol. 23, pp. 617–622, June 2007.
- [10] P. McDowell, J. Chen, and B. Bourgeois, "UUV teams, control from a biological perspective," in *IEEE Oceans Conference Rec.*, vol. 1, (Mississippi, MS), pp. 331–337, 2002.
- [11] R. T. Duan M. Stipanovic, Gkhan Inalhan and C. J. Tomlin, "Decentralized overlapping control of a formation of unmanned aerial vehicles," *IEEE Conference on Decision and Control*, p. 28292835, 2002.
- [12] M. Anderson and A. Robbins, "Formation flight as a cooperative game," in *AIAA Guidance, Navigation and Control Conference*, (Boston, MA), pp. 244–251, 1998.
- [13] T. Balch and R. Arkin, "Behavior-based formation control for multirobot teams," *IEEE Transactions on Robotics and Automation*, vol. 14, no. 6, pp. 926–936.
- [14] F. Fahimi and D. Schoerling, "Sliding-mode formation controller for an underactuated surface vessel. part ii/ii,"
- [15] P. Encarnacao and A. Pascoal, "Combined trajectory tracking and path following: an application to the coordinated control of autonomous marine craft," in *Proceedings of the 40th IEEE Conference on Decision and Control*, pp. 964–969, December 2001.
- [16] A. Behal, D.M.Dawson, B.Xian, and P.Setlur, "Adaptive tracking control of underactuated surface vessels," in *Proceeding of the 2001 IEEE International Conference on Control Applications*, pp. 645–650, September 2001.

- [17] K. Pettersen and H. Nijmeijer, "Tracking control of an underactuated surface vessel," in *Proceedings of the 37th IEEE Conference on Decision and Control*, pp. 4561–4566, December 1998.
- [18] M. Rayhanoglu, "Exponential stabilization of an underactuated autonomous surface vessel?," *Automatica*, vol. 33, no. 12, pp. 2249–2254, 1997.
- [19] F. Fahimi, S. Rineesh, and C. Nataraj, "Formation controllers for underactuated surface vessels and zero-dynamics stability," *International Journal of Intelligent Systems and Control*, pp. 1–11, 2008. in print.
- [20] S. Sheikholeslam and C. Desoer, "Control of interconnected nonlinear dynamical systems: The platoon problem," *IEEE Transactions on Automatic Control*, vol. 37, no. 6, pp. 806–810.
- [21] J.-J. E. Slotine and W. Li, *Applied nonlinear control*. Upper Saddle River, New Jersey, USA: Prentice Hall, 1991.
- [22] Rotomotion, *VTOL AFCS. Reference Manual*. Rotomotion, LLC, 501C La Mesa St. Mt. Pleasant, SC 29464, USA, 1.0 ed., 2003. Part Number: DO-CAFCS24REFMAN001.
- [23] Microrobotics, *Servo Switch/Controller. Users Manual*. Microrobotics, Inc., 28 Research Drive, Suite G, Hampton, Virginia, USA, 2007.
- [24] Advanced Digital Logic, *ADL855PC Manual*. Advanced Digital Logic, Inc, 4411 Morena Blvd., Suite 101, San Diego, CA, 2.0 ed.
- [25] D. Schoerling, "Design and test of a robust formation controller for an unmanned surface vessel," Master's thesis, University of Alberta, Technische Universitat Bergakademie Freiberg, 2009.

APPENDIX A

EMBEDDED SOURCE CODE

```
function qpp = DynamicEqns(speed, angle, const, q, qp)
% This block supports an embeddable subset of the MATLAB language.
% See the help menu for details.

m11=const(1); %redeclare constant parameters
m22=const(2);
m33=const(3);
d11=const(4);
d22=const(5);
d66=const(6);

xp=qp(1); % recelare the state terms x', y', theta'
yp=qp(2);
thetap=qp(3);
theta=q(3); % also redeclare the state paramter theta

% define the local velocities
u=xp*cos(theta) + yp*sin(theta);
```

```

v=-xp*sin(theta) + yp*cos(theta);

% calculate forces from inputs (speed, angle)
D = 0.01; %prop diameter, in meters
L = 0.25; % distance from prop to cg, in meters
p = 998; % density of water, in kg/m3
Kt = 0.14; % propellor thrust coeficient
%Va = u; % advance speed of water prior to passing through the prop.
th = p*(D4)*Kt*abs(speed)*speed; % thrust(per propeller)

X = th*cos(angle); % surge force
Y = th*sin(angle); % sway force
N = -L*Y; % yaw torque

% accelerations = outputs
xpp = cos(theta)*(((m22-m11)/m11)*v*thetap - (d11/m11)*u +X/m11) - sin(theta)*(((m22-
m11)/m22)*u*thetap -(d22/m22)*v +Y/m22);
ypp = sin(theta)*(((m22-m11)/m11)*v*thetap - (d11/m11)*u +X/m11) + cos(theta)*(((m22-
m11)/m22)*u*thetap -(d22/m22)*v +Y/m22);
thetapp = (1/m33)*(-(m22-m11)*u*v -d66*thetap + N);
qpp = [xpp; ypp; thetapp];

```

APPENDIX B

CONTROLLER BOX: CONSTRUCTION PROCEDURE

Before building the box review the parts list, as shown in "Documentation.doc.", to confirm all parts are available.

Steps 1 through 13 can be done before wiring, but be sure to have read and completed wiring (3-way switch, battery, and servo wires) before starting steps 14 through 16.

Also some parts were built from scratch by the machine shop: " BACKPLATE " L-BRACKETS " TOPPLATE " ENCLOSURE BOLTS (not shown in model)

Other parts will need to be modified by the machine shop:

- ENCLOSURE_2 - Hole for the GPS antennae** - Shave down corner blind holes
- Drill out holes for THREADED_INSERTS - ENCLOSURE_TOP3 - Cut the ENCLOSURE_TOP3 in half - Glue TOPPLATE to ENCLOSURE_TOP3 - Drill holes for SERVO_CONN, SWITCH, and LED (2)

1. Insert PC104 into BACKPLATE; tighten with NUT_LARGE.
2. Have machine shop weld the 4 L-brackets together, thread STANDOFF_SMALL_SHORT into L-BRACKET_FRONT, L-BRACKET_LEFT, and L-BRACKET_RIGHT. Also thread STANDOFF_SMALL into L-BRACKET_BACK.
3. Connect L-brackets with BACKPLATE by inserting 4 SCREW_LARGE_LONG

and then tighten with NUT_LARGE.

4. Align PCB_1 with L-BRACKET_RIGHT, then screw in with 4 SCREW_SMALL_SHORT.
5. Align PCB_2 with L-BRACKET_FRONT, then screw in with 4 SCREW_SMALL_SHORT.
6. Align PCB_3 with L-BRACKET_LEFT, then screw in with 3 SCREW_SMALL_SHORT.
7. Align PCB_4 with the right side of L-BRACKET_BACK, then screw in with 4 STANDOFF_SMALL.
8. Place 4 THREADED_INSERT into blind holes in the ENCLOSURE.
9. Place BACKPLATE into the ENCLOSURE, then screw in 4 SCREW_BACKPLATE.
10. Screw GPS_HEATSINK into L-BRACKET_BACK, with SCREW_SMALL.
11. Insert GPS antennae into the hole in the ENCLOSURE, then align with L-BRACKET_BACK, then screw in 4 STANDOFF_SMALL. ****Important notice****
12. Screw in 4 STANDOFF_LARGE_SHORT into HDD, then align with TOPPLATE and screw in 4 SCREW_LARGE_LONG.
13. Align 4 STANDOFF_SMALL_FEMALE with SERVO_BOARD, then screw in 4 SCREW_SMALL_LONG. Then, align with TOPPLATE, and screw in 4 more SCREW_SMALL_LONG.

Be sure to complete wiring for battery, the 3-way switch, and the servo input/output

14. Screw the Velcro pieces into the remaining holes of the TOPPLATE, with SCREW_LARGE_LONG and NUT_LARGE. Be sure to place WASHER_LARGE between the Velcro and the screw head. Also, glue two "loop" strips of Velcro to the BATTERY Then wrap the Velcro around the battery from both sides to secure it in place.
15. After wiring has been set-up for the servo input/output, place SERVO_CONN into slot in the ENCLOSURE_TOP3.
16. After wiring has been done for the three-way switch, insert the SWITCH into the

ENCLOSURE_TOP3, and fasten the nut.

** The hole in the ENCLOSURE for the GPS antennae connector has not been explicitly designed. Once the manufacturing of the parts for the box have been finished, line up the GPS with the STANDOFF_SMALL_LONG and mark where the hole should be. Then proceed to drill a hole for the antennae connector.

APPENDIX C

HARDWARE AND SOFTWARE LISTS

The following hardware was used during the course of this research. Many of these items are covered in the experimental setup chapter. They are listed here with the software programs used as an easy to look up reference.

- Sensor. From Rotomotion, the AHPRS200A, 200Hz Attitude, Heading and Position Reference System is used to get the boat state. For more info and ordering refer to www.rotomotion.com.
- SSC. The Servo Switch/Controller from Microbotics, Inc. is used to connect the computer to the servos and the radio receiver. Allowing switching between manual and computer control of servos/motors. Refer to www.microboticsinc.com/rc_servo_controller.html for more information.
- PC/104. The onboard computer system used is a ADL855PC from Advanced Digital Logic. Go to www.adl-usa.com for more information. This computer runs the controller and virtual leader boat using an embedded 'real time workshop' executable on a xpctarget kernel, both from MATLAB.
- Boat. The boat used is a Villian racer from Traxxas. The hull and the steering servo and motors with speed controller are used, but no the radio receiver or the dorsal cover plate.

- Transmitter and receiver. To manually operate the boat and to trigger the computer control a XR3i radio transmitter from JR Racing was used. This in conjunction with a 3 channel racing receiver of model R135 also from JR Racing. Refer to www.jrradios.com for further detail.
- Power supply. A 12 V power supply was used for testing, debugging of the system to provide a steady current and power level and reduce/prevent draining batteries. The PSU used was a Mean Well model SP-100-12, refer to www.meanwelldirect.co.uk for further specification.
- Li-Po batteries. For water testing, or other outdoor procedures the electronics are powered by Lithium-polymer batteries from Parker.
- NiMH batteries. For powering the boat motors Nickel metal hydride batteries are used. These are DTXC2146 from DuraTrax. See www.duratrax.com.
- For rapid transference of the 'RTW' files from the host PC to the PC/104 a USB 2.0 Drive mate was used to connect the onboard hard drive to the host PC. This interface dongle is from BYTECC. Refer to www.byteccusa.com/ for more details.
- Additionally a digital multi meter and an oscilloscope were used for debugging and troubleshooting the hardware setup.

During this research the following software programs were used.

- MATLAB. MATLAB was used exhaustively for the research. All programming for simulations of the boat and controller were done with the Simulink toolbox. Further, the onboard programs, were compiled based on the Simulink models and with the xpctarget tool-set. The xpctarget tools allow the user to run models on target computers with the xpctarget kernel, and allow host to target

communication between computers. The basic MATLAB environment was used for all data processing and production of figures. MATLAB can be obtained from www.mathworks.com/ where there is also an intensive help file system and support community available.

- **WATCOM.** This is not so much a program as a compiler (and libraries) that was used for the compilation of the executable programs for the onboard computer. Note that the version of Watcom used depends on the version of MATLAB used as not all versions have intercompatibility. Watcom, as open source freeware, can be obtained at www.openwatcom.org/. Refer to MATLAB help files for setting Watcom as the default compiler for xpctarget applications.
- **Ground.** This program, packaged with the Rotomotion Hardware, runs a GUI that allows the user to interact and view the outputs of the sensor. Ground also allows some user inputs to configure the sensor such as re-calibration of the magnetometer. For further information about this program one can look at the source code online at <http://autopilot.cvs.sourceforge.net/viewvc/autopilot/>, or contact Rotomotion at the www.rotomotion.com for further details.
- **Wireshark.** This program allows the user to packet sniff ethernet connections. Specifically useful for troubleshooting the TCP and UDP communications between the sensor and PC/104, between target and host computers, and sensor and host. Wireshark is a free software program and can be obtained at www.wireshark.org/ along with user instructions.
- **HyperTerminal.** HyperTerminal is a small application that enables connection to remote devices. The use in this research was to connect to the SSC for configuring the switch table and message outputs. This program comes with

all recent versions of the Windows OS, for others there are version available online, along with user manuals and tutorials. The connection settings used are described in the SSC manual.