2012

Incorporating input saturation for underactuated surface vessel trajectory tracking control

Yaswanth Siramdasu

INCORPORATING INPUT SATURATION FOR UNDERACTUATED SURFACE VESSEL TRAJECTORY TRACKING CONTROL

by

YASWANTH SIRAMDASU

A THESIS

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Engineering
in
The Department of Mechanical and Aerospace Engineering
to
The School of Graduate Studies
of
The University of Alabama in Huntsville

HUNTSVILLE, ALABAMA

2012

In presenting this thesis in partial fulfillment of the requirements for a master's degree from The University of Alabama in Huntsville, I agree that the Library of this University shall make it freely available for inspection. I further agree that permission for extensive copying for scholarly purposes may be granted by my advisor or, in his/her absence, by the Chair of the Department or the Dean of the School of Graduate Studies. It is also understood that due recognition shall be given to me and to The University of Alabama in Huntsville in any scholarly use which may be made of any material in this thesis.

Yaswanth Siramdasu — 05-24-2012

Yaswanth Siramdasu — (date)

THESIS APPROVAL FORM

Submitted by Yaswanth Siramdasu in partial fulfillment of the requirements for the degree of Master of Science in Engineering in Mechanical Engineering and accepted on behalf of the Faculty of the School of Graduate Studies by the thesis committee.

We, the undersigned members of the Graduate Faculty of The University of Alabama in Huntsville, certify that we have advised and/or supervised the candidate of the work described in this thesis. We further certify that we have reviewed the thesis manuscript and approve it in partial fulfillment of the requirements for the degree of Master of Science in Engineering in Mechanical Engineering.

5-21-12
Dr. Farbod Fahimi (Date) Committee Chair

5-21-12
Dr. Nathan J. Siegers (Date)

5/21/12
Dr. D. Brian Landrum (Date)

6/22/12 Department Chair
Dr. Keith Hollingsworth (Date)

06/25/12 College Dean
Dr. Shankar Mahalingam (Date)

7/10/12 Graduate Dean
Dr. Rhonda Kay Gaede (Date)

ABSTRACT

School of Graduate Studies
The University of Alabama in Huntsville

Degree Master of Science in Engineering College/Dept. Engineering/Mechanical and Aerospace Engineering

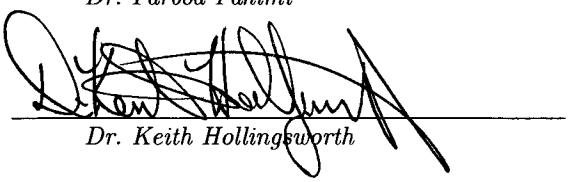
Name of Candidate Yaswanth Siramdasu

Title Incorporating Input Saturation for Underactuated Surface Vessel Trajectory Tracking Control

A Nonlinear Model Predictive Controller (NMPC) for trajectory tracking of surface vessels which takes care of input saturation is designed based on a 3DOF dynamic model. NMPC calculates the future control inputs (propeller speed and rudder angle of the vessel) based on the present state variables by optimizing a cost function. The cost function incorporates input constraints as well as state errors in determining the control inputs is exploited. The identification tests are performed in an outdoor pool on a robotic boat which has an onboard PC104 computer and a navigation sensor. The data from experiments are used to determine the six model parameters. A NMPC is designed based on the identified model to track a sine wave. The performance of the controller is demonstrated through simulations via MATLABs Simulink program and tested on a real robotic boat. The experimental results show for input saturation case the control inputs remain within the saturation limits in extreme maneuvers, the vessel recovers from saturation, and the vessel follows the trajectory very closely when the inputs are not saturated.

Abstract Approval: Committee Chair 

Dr. Farbod Fahimi

Department Chair 

Dr. Keith Hollingsworth

Graduate Dean 

Dr. Rhonda Kay Gaede

ACKNOWLEDGMENTS

Without the help of my advisor Dr. Farbod Fahimi, I could never accomplish the study towards a M.S degree. His experienced comments and advices from the higher levels always bring me a fresh thinking and passion towards my work. These instructions are the most important sections of what I have learned during my study. I am very much thankful to him for spending uncountable hours with me for discussions and for his full support at my surgery time. I would also like to thank my lab mates Joshua Hill, Richard Dyar and Chandra Sekhara Bharath for all their help and fun in lab. I would like to thank Dr. Nathan Slegers and Dr. Brian Landrum for being as committee members. I also thank my friends at Overlook who made my stay in US to feel at home for taking care of me at my surgery time. I thank all those who directly or indirectly influenced me all through my life. Last but not least is my parents for their invaluable support and sacrifices.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

TABLE PAGE

Dedicated to my grandma, the most loving person to me on earth and aspired me all through my life.

CHAPTER 1

INTRODUCTION

The use of Surface Vessels in marine applications has grown over the past decade. The important applications include surveillance and supporting minesweeping operations. For these applications, surface vessels have to be operated for long periods of time. Using human operators remotely control and monitor the vessels for long time spans negates the advantage of using robotic surface vessels completely. Hence designing an automated surface vessel with a controller which reduces or eliminates the human interactions is desired. However, developing a control algorithm that can accomodate disturbances, model uncertainties and other constraints on system to follow the desired path is challenging [1].

Controlling the path of vessel on the desired path is done in two ways. One is way-point control, where a series of desired locations and desired orientations of the surface vessel are defined. The way-point controller then guides the vessel such that it meets these desired locations and orientations. The path of the vessel between one way-point to the next way-point is not defined and it follows an arbitrary path based on the control inputs which meet the next way-point. Researchers have developed

way-point controllers based on different control strategies [2], [3], [4]. The other controller strategy, trajectory tracking control is fundamentally different than way-point controller is the approach presented in this thesis. Trajectory tracking control unlike way-point control, controls the vessel to follow a time-dependent speed profile curve such that vessel meets the desired position and orientation at every point in time.

Developing a trajectory tracking control for surface vessels is challenging due to their underactuated dynamics. Usually for surface vessels a propeller and rudder are used as controlling inputs, i.e the number of DOFs in a planar motion (3) is higher than the number of control inputs (2). The absence of any kinematic constraints among the degrees of freedom make them holonomic in nature (there is no control in the lateral direction). Therefore, developing a optimal trajectory tracking controller for underactuated surface vessels of significant research interest. Over the past researchers developed various linear and nonlinear trajectory tracking controls listed in Table 1.1.

All the cited references controlled the two positions of center of gravity (CG) of the vessel and incorporated another sub controller to control orientation. Here a novel way of controlling the control point (CP) of underactuated surface vessel is applied in designing the controller. The CP is a point on vessel other than CG whose position is a function of CG and orientation [25].

Table 1.1: Review of unmanned surface vessel control

Ref.	Approach	Method	Test	Yr
[5]	Linear	LQG feedback	Outdoor	96
[6]	Linear	LQG feedback	Outdoor	97
[7]	Linear	LQG feedback	No	98
	Linear	Cascade control	No	98
[8]	Nonlinear	Backstepping	No	98
[9]	Linear	Backstepping	No	99
[10]	Nonlinear	Backstepping	Indoor	01
[11]	Nonlinear	Backstepping	No	03
[12]	Nonlinear	Fuzzy control	No	03
[13]	Nonlinear	Backstepping	No	03
[14]	Linear	Cascade control	Indoor	03
[15]	Nonlinear	Cascade control	Indoor	04
[16]	Nonlinear	PI-type sliding mode	No	04
[17]	Nonlinear	State feedback	No	05
[18]	Nonlinear	Backstepping	No	05
[19]	Nonlinear	Neural network	No	05
[20]	Nonlinear	Backstepping	No	06
[21]	Nonlinear	Fuzzy control	No	07
[22]	Nonlinear	Sliding mode	No	07
[23]	Nonlinear	Sliding mode	Indoor	08
[24]	Nonlinear	PID vs. fuzzy	No	09

Here in this thesis another practical problem of input saturation of underactuated surface vessels is addressed. Input saturation is the common practical problem in all dynamic systems which makes the system to overshoot and oscillate and thus degrading the performance of the system or in some cases due to disturbances or large overshoot turns the system instable. Over the past, researches addressed this problem and developed control algorithms for linear and nonlinear systems with input saturation. Huang [26] used fuzzy PD to tunes the proportional gains dynamically and Li [27] used a variable-gain saturation function to the main controller which has high gain when the control input is low and lower gains when it approaches higher control

inputs, these gain scheduling methods guaranteed closed loop stability also. Wen [28] designed a controller using backstepping approach and included a separate Nussbaum function which takes care of nonlinear terms causing input saturation. Garelli [29] used a variable structure with a sliding regime, which is defined as difference in maximum control input to actual control input. The control input to the plant is applied until the asymptotical stability of this sliding regime is achieved which makes the difference equal to zero at the saturation.

Several control laws without or with modification to the main control law to include input saturation are applied to aircraft control [30], automotive vehicle control [31, 32], manipulators path control [33, 34] but input saturation effects in surface vessels has not been addressed much. Li Junfang [35] proposed a Neutral Network algorithm for the ship course control with input saturation limits on the rudder angle which is not a trajectory tracking. Do.K.D [36] used Lyapunov's direct method and backstepping technique to develop a controller for stabilizing and tracking of underactuated ships in which it is mentioned that their method could be extended to include input saturation. However, such an extension has not been discussed.

Trajectory tracking controller for surface vessels with input constraints is difficult due to its underactuated dynamics of the surface vessels and the controller can not guarantee the stability of the system under saturation. Although, imposing inputs saturation using a simple 'if' logical statement constraints the inputs but it can not guarantee the closed loop stability and it is demonstrated through the following

example. An initial offset in between actual CP of the boat and desired CP is specified to make the control inputs saturate. As an illustrating example Figure 1.1 shows the controller calculated a negative thrust and a rudder angle large than the maximum which can not be attained in the real physical boat. Figure 1.2 shows propeller speed is constrained to $0.03|F_{max}|$ and rudder to the maximum value, using a 'if' logical statement, until the boat converges to the desired path, so that it will not overshoot or stay behind. The value 0.03 is selected based on trial and error method so that it makes the boat stable in this particular situation in simulations using a sliding mode controller. However, in real time application these will not be the exact conditions of operation of the boat. If the value is too small the boat is left behind or if the values is too high the boat overshoots the desired path and finally becomes unstable.

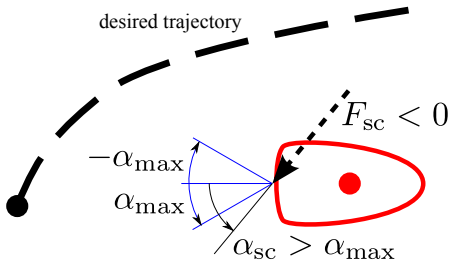


Figure 1.1: Calculated control inputs without saturation limits

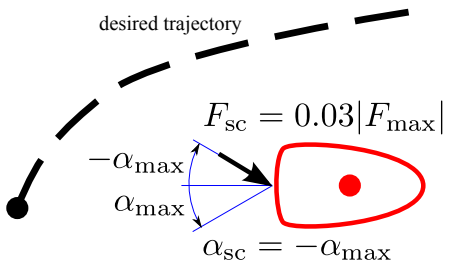


Figure 1.2: Calculated control inputs with defined saturation limits

Recently, a trajectory-tracking controller was developed for underactuated surface vessels using the sliding mode control method and was successfully tested on a robotic boat [37]. The boat's control inputs were the propeller speed and the rudder angle, both of which had some saturation limits. During the tests reported in [37], it

was discovered that if the boat was too far from the desired trajectory at the beginning of the test, the controller would demand high inputs that surpass the saturation limits. The saturated inputs showed extensive chattering, which led to instability and the tests would fail. At that time, the problem was solved by initializing the control with a PID way-point controller that drove the boat to the start point of the desired trajectory, where the sliding mode controller took over. However, since the same situation could arise if excessive disturbance moves the boat too far from its desired trajectory at any time during tracking. So a controller which is aware of its path beforehand and makes real time online calculations for the convergence of the solution, considering the constraints, must be designed.

Model Predictive Control (MPC) is a good tool for addressing this input saturation problems. MPC is real time dynamic optimization tool, which predicts the future states by optimizing a cost function. The cost function contains an error function in states vector and a function defining the constraints on the system like input constraints [38] or economical constraints [39] or environmental constraints [40]. The main advantage of Model Predictive Control (MPC) is almost all practical problems of the system can be incorporated into the cost function to solve for control input which considers all constraints, achieving all these with other existing controls strategies in a very difficult task. Model Predictive Control was first used in process control industries which has slow varying variables with large sampling time. Now a days with the advent of fast computation power and development of optimal algorithms tools made its way into almost all applications include aerospace [41], manufacturing

industries [42], automotive [43], chemical [44], nuclear power plants [45], structural mechanics [46] and in other numerous fields. Its present and future directions [47] and application can be found in detail [48].

Trajectory Tracking Model Predictive Controller was studied on rotorcrafts [49] and ground vehicles [50] but it was not applied for the control of underactuated surface vessels. Although, Zhen Li [51] developed MPC control with rudder and roll constraints for surface vessel but it is for path following unlike trajectory tracking it does not have to meet the temporal specifications. For the first time, a Nonlinear Model Predictive Controller (NMPC) for the trajectory tracking of underactuated surface vessel which is aware of input constraints is designed and tested in this thesis. The dynamic model is very important in NMPC design because the decisions are taken based on this model. Identification tests are performed and an accurate dynamic model is estimated.

CHAPTER 2

DYNAMIC MODEL

The robotic boat used in this thesis, which has a 6 DOF is displayed in Figure 2.1. These 6 DOFs constitute x, y, z positions of the center mass of the boat and respective angles which define orientation of the boat with respect to global frame of reference. The rate of change of these six DOFs constitute six generalized velocity components, which define the dynamic state of the vessel at any time. The generalized velocities in the vessel's body frame of reference are longitudinal (u), transverse (v) and normal (w) linear speeds, and angular velocities about longitudinal (p), transversal (q), and normal(r) axes. The 6 DOF dynamic model of the surface vessel simulated the motion of the boat. For surface vessels usually there are two control inputs. Here two independent propellers provide driving forces for linear motion and steering torque for the rotation. The Eq.s (2.1) to (2.6) representing the dynamics of the vessel in local coordinate system are derived based on following assumptions: the inertia of the vessel is assumed to be constant, the vessel is closely represented by elliptical body and the higher-order damping forces are neglected.

Figure 2.1: Robotic boat with control box and GPS antenna

$$m_{11}\dot{u} - m_{22}vr + m_{33}wq + d_{11}u = W_u + F, \tag{2.1}$$

$$m_{22}\dot{v} - m_{33}wp + m_{11}ur + d_{22}v = W_v, \tag{2.2}$$

$$m_{33}\dot{w} - m_{11}uq + m_{22}vp + d_{33}w = W_w + mg + Z_w, \tag{2.3}$$

$$I_{xx}\dot{p} + (m_{33} - m_{22})vw + (I_{xx} - I_{yy})rq + d_{44}p = K_p, \tag{2.4}$$

$$I_{yy}\dot{q} + (m_{11} - m_{33})uw + (I_{xx} - I_{zz})rp + d_{55}q = F, \bar{F}G + M_q, \tag{2.5}$$

$$I_{zz}\dot{r} + (m_{22} - m_{11})uv + (I_{zz} - I_{yy})qp + d_{33}r = T. \tag{2.6}$$

where m_{ij}'s are effective mass of the vessel which includes mass of the vessel plus mass of the water carrying with it (due to the effect of hydrodynamics), d_{ij}'s are the linear hydrodynamic damping coefficients, I_{ii}'s are effective moment of inertia, and W_i are the unknown wave forces on vessel described in terms of the vessel's local frame. Z_w is the buoyancy force acting upwards. $F, \bar{FG}$ is the generated torque from the thruster's force acting about the transversal (pitch) axis passing through the vessel's center of mass. K_p and M_q are roll and pitch restoring toques given by

$$K_p = -mgMT_p \sin\phi, \quad M_q = -mgMT_q \sin\theta, \quad Z_w = -\rho g A_{wp} z. \tag{2.7}$$

where MT_p and MT_q are the transverse and longitudinal metacentric heights and A_{wp} denots the water planar area. ϕ and θ are the roll and pitch angles of the vessel's local frame.

F is the propeller force which drives the vessel along longitudinal direction. T is the torque generated from propeller and rudder of the vessel, which causes the rotation of vessel along z axis (yaw motion) Figure 2.2. These two are the controlling inputs of the vessel.

2.1 Simplified Dynamic Model

From Eq.s (2.1) to (2.6), the relation of forces and torques on the vessel in all three directions to two control inputs is very complicated. The 6 DOF dynamic model can be reduced to 3 DOF model by taking the advantage of inherent stability of

some of the DOF's of the vessel, which is intentionally built into vessel's dynamics at the designing stage for a good vessel design. The buoyancy force stabilizes the heave motion i.e linear motion along z axes. The longitudinal and transversal metacentric heights of the vessel are designed to be above the CG of the vessel, which stabilizes the roll and pitch motions of the vessel. The stability of heave (w), roll (p) and pitch (q) motions and the assumption of small angular DOFs reduce the 6 DOF to 3 DOF model of the surface vessel given by

$$\begin{aligned} m_{11}\dot{u} - m_{22}vr + d_{11}u &= W_u + F, \\ m_{22}\dot{v} + m_{11}ur + d_{22}v &= W_v, \\ I_{33}\dot{r} + (m_{22} - m_{11})uv + d_{66}r &= T. \end{aligned} \tag{2.8}$$

where u (longitudinal velocity), v (lateral velocity) and r (yaw rate) represents in vessel's body frame Figure 2.2. Note the second equation is not directly affected by the control inputs and this equation represents the lateral response of the vessel. Thus, there is no control over the lateral motion of the vessel. This corresponds to the underactuated nature of the surface vessel.

2.1.1 Role of Control Point

For designing a controller for the vessel, the number of control inputs should be equal to the DOFs of the vessel. Hence out of 3 DOFs most important 2 DOFs has to be selected which can be controlled using two control inputs. In trajectory tracking, the two linear position components of the center of gravity of the vessel are

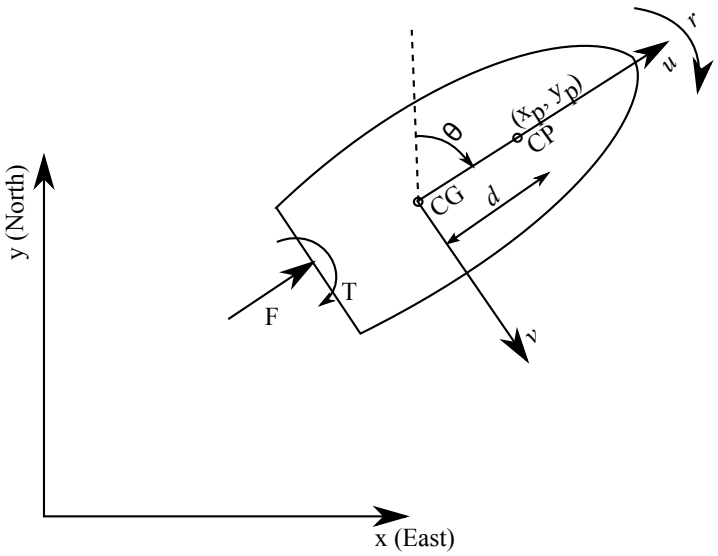


Figure 2.2: Global frame of reference and local frame of reference of the boat

most important for the better tracking performance. The disturbance in other DOF, orientation of the vessel is not sensed by the controller. This approach works well in theory and in practice under low disturbances. This is because the longitudinal hydrodynamic forces inherently stabilize vessel while its moving. But in practice, any high disturbance, the disturbance from the water currents, wave, etc., disturb the orientation of the vessel. Since, the controller only feeds back the position of CG, it cannot directly react to change in orientation. In extreme cases, the longitudinal hydrodynamic forces converges it to unstable state and ending up going backward (180° shift in orientation) while its CG following the trajectory.

Rather than controlling the position components of CG of the vessel, controlling a point on the vessel's body other than that of CG can solve this problem. A point called "control point", is selected arbitrarily on the vessel other than its center

of gravity, whose position is controlled for tracking a trajectory. This concept has been applied successfully to nonholonomic mobile robots in the robotics literature. However, comparing the kinematics, dynamics, and work environment of underactuated surface vessels with that of mobile robots is not appropriate by any stretch of imagination, until now there has not been any proof that this concept would work for underactuated surface vessels. In this research, for the first time, the idea of using a control point for trajectory tracking control of surface vessels is theoretically and experimentally investigated.

The position components of control point (Eqs. (2.9) and (2.10)) are functions of all DOFs of the vessel. A disturbance generated in the orientation of the vessel generates an output error in position of control point, to which the controller can react. This controller action gives better tracking performance which is not the case of controlling CG of the vessel. Selecting the control point on the positive longitudinal axis of the surface vessel with a distance of d from the CG simplifies the relation between the inputs and outputs of the controller.

$$x_p = x + d\sin\theta, \tag{2.9}$$

$$y_p = y + d\cos\theta. \tag{2.10}$$

where x_p and y_p indicate the global position components of the vessel's control point, and θ is the vessel's orientation (Figure 2.2).

The two physical control inputs for the vessel are propeller speed (n) and rudder angle

(α). The dynamic equations expressed in boat's frame of reference (Figure 2.2), with these control inputs and control point as controller output and assuming the disturbance to be small, Eqs. (2.8) condenses to

$$m_{11}\dot{u} - m_{22}vr + d_{11}u = F\cos\alpha, \tag{2.11}$$

$$m_{22}\dot{v} + m_{11}ur + d_{22}v = F\sin\alpha, \tag{2.12}$$

$$I_{33}\dot{r} + (m_{22} - m_{11})uv + d_{66}r = -(F\sin\alpha)L. \tag{2.13}$$

2.2 Configuration of Dynamic Model into State Space

The global velocity of the control point is (Figure 2.2)

$$\dot{x}_p = \dot{x} + rd\cos\theta, \quad \dot{y}_p = \dot{y} - rd\sin\theta, \tag{2.14}$$

where

$$\dot{x} = u\sin\theta + v\cos\theta, \quad \dot{y} = u\cos\theta - v\sin\theta. \tag{2.15}$$

A configuration vector $\mathbf{x} = [x_p, y_p, \theta]^T$ is defined. The velocity components of the vessel's center of gravity in its local frame (u, v) and its yaw rate ($r = \dot{\theta}$) are arranged as $\mathbf{v} = [u, v, r]^T$. The kinematic equations that convert the local speeds to global speeds and the equations of motion (2.11) to (2.13) are converted to a matrix form for easier manipulation.

The kinematic equations relating velocity components in the global frame Eq. (2.14) to those in the local frame $\mathbf{v}$ and the dynamic equations of motion in local frame of the vessel (Eqs. (2.11) to (2.13)) are represented in matrix form as

$$\dot{\mathbf{x}} = \mathbf{R}(\theta)\mathbf{v}, \tag{2.16}$$

$$\mathbf{M}\dot{\mathbf{v}} + \mathbf{C}(\mathbf{v})\mathbf{v} + \mathbf{D}\mathbf{v} = \mathbf{F}. \tag{2.17}$$

The matrix $\mathbf{R}(\theta)$ is conversation matrix which converts the local velocities to global velocities.

$$\mathbf{R}(\theta) = \begin{bmatrix} \sin\theta & \cos\theta & d\cos\theta \\ \cos\theta & -\sin\theta & -d\sin\theta \\ 0 & 0 & 1 \end{bmatrix}. \tag{2.18}$$

where mass matrix $\mathbf{M} = \text{diag}(m_{11}, m_{22}, I_{33})$ and damping coefficient matrix $\mathbf{D} = \text{diag}(d_{11}, d_{22}, d_{66})$. The matrix $\mathbf{C}(\mathbf{v})$ contains inertial forces which causes Coriolis acceleration,

$$\mathbf{C}(\mathbf{v}) = \begin{bmatrix} 0 & 0 & -m_{22}v \\ 0 & 0 & m_{11}u \\ m_{22}v & -m_{11}u & 0 \end{bmatrix}. \tag{2.19}$$

The vector $\mathbf{F_n} = [F\cos\alpha, F\sin\alpha, -FL\sin\alpha]^T$ contains the longitudinal and lateral components of the propeller force (F), and the propeller's moment about the vessel's CG, which has a longitudinal offset L with the propeller location on the vessel. The propeller force is modeled as $F = A|n|^C$, where A is a constant and C is

a dimensionless parameter which relates the thrust on vessel to propeller speed. The propeller force can be written as

$$\mathbf{F} = [AU_1, AU_2, -ALU_2]^T, \tag{2.20}$$

where

$$U_1 = |n|^C \cos\alpha \quad U_2 = |n|^C \sin\alpha \tag{2.21}$$

are selected as mathematical control inputs. Now, $\mathbf{F}$ can be separated and put in a matrix multiplication as,

$$\mathbf{F} = \begin{bmatrix} A & 0 \\ 0 & A \\ 0 & -AL \end{bmatrix} \begin{bmatrix} U_1 \\ U_2 \end{bmatrix} = \mathbf{Au}. \tag{2.22}$$

Finally, Eqs. (2.17) and (2.16) can be written as

$$\dot{\mathbf{x}} = \mathbf{R}(\theta)\mathbf{v}, \tag{2.23}$$

$$\dot{\mathbf{v}} = \mathbf{M}^{-1}(\mathbf{Au} - (\mathbf{C}(\mathbf{v}) + \mathbf{D})\mathbf{v}). \tag{2.24}$$

The complete dynamic model of the surface vessel is obtained by combining Eqs. (2.23) and (2.24) given by,

$$\begin{bmatrix} \dot{x_p} \\ \dot{y_p} \\ \dot{\theta} \\ \dot{u} \\ \dot{v} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} u\sin\theta + (v + rd)\cos\theta \\ u\cos\theta - (v + rd)\sin\theta \\ r \\ \frac{1}{m_{11}}(AU_1 - d_{11}u + m_{22}vr) \\ \frac{1}{m_{22}}(AU_2 - m_{11}ur - d_{22}v) \\ \frac{1}{I_{33}}((m_{11} - m_{22})uv - LAU_2 - d_{66}r) \end{bmatrix} \tag{2.25}$$

$$\begin{bmatrix} \dot{q_1} \\ \dot{q_2} \\ \dot{q_3} \\ \dot{q_4} \\ \dot{q_5} \\ \dot{q_6} \end{bmatrix} = \begin{bmatrix} q_4\sin q_3 + (q_5 + q_6 d)\cos q_3 \\ q_4\cos q_3 - (q_5 + q_6 d)\sin q_3 \\ q_6 \\ \frac{1}{m_{11}}(AU_1 - d_{11}q_4 + m_{22}q_5 q_6) \\ \frac{1}{m_{22}}(AU_2 - m_{11}q_4 q_6 - d_{22}q_5) \\ \frac{1}{I_{33}}((m_{11} - m_{22})q_4 q_5 - LAU_2 - d_{66}q_6) \end{bmatrix} \tag{2.26}$$

$$\dot{\mathbf{q}} = \mathbf{f}(\mathbf{q}, \mathbf{U}), \tag{2.27}$$

where $\mathbf{q} = [x_p, y_p, \theta, u, v, r]^T$.

CHAPTER 3

MODEL PARAMETER IDENTIFICATION

A robotic small boat is designed and built for control experimentation purposes (Figure 2.1). The specifications of the boat are length 80 cm, width 70 cm, weighing mass 8 Kg, diameter of each propeller 4.5 cm respectively. It is integrated with a control box which has an embedded computer PC-104 form factor with a 1.8 GHz Pentium M processor and a navigation sensor. The navigation sensor gives vessel's state vector values and a serial port communicates these values to PC-104. A Servo Switch Card (SSC) sends the signals from PC-104 to the respective servo motors on the boat. The navigational sensor consists of a GPS unit which outputs (x, y, z) position components, 3-axes accelerometers and gyroscopes gives roll, pitch, yaw Euler angles. Using these 6 values, state space vector $\mathbf{q} = [x_p, y_p, \theta, u, v, r]^T$ is fedback to the controller to estimate the desired control inputs. Open-loop identification test programs are coded using Matlab/Simulink and xPC-Target rapid control prototyping software. The software runs in real-time on the embedded computer. Using xPC-Target open-loop inputs and the states of the vessels are recorded in real-time for identification. Detailed description about the hardware setup is presented in Appendix.

3.1 Test Scenarios

The components of Eq. (2.20) are combined with Eq. (2.11) to formulate a form of equations of motion that is suitable for identification.

$$(m_{11}/A)\dot{u} - (m_{22}/A)vr + (d_{11}/A)u = U_1, \tag{3.1}$$

$$(m_{22}/A)\dot{v} + (m_{11}/A)ur + (d_{22}/A)v = U_2, \tag{3.2}$$

$$(I_{33}/A)\dot{r} + ((m_{22} - m_{11})/A)uv + (d_{66}/A)r = -LU_2. \tag{3.3}$$

By observing this form, one can see that six parameters m_{11}/A, m_{22}/A, m_{33}/A, d_{11}/A, d_{22}/A, and d_{66}/A in Eq.s (3.1) to (3.3) and dimensionless parameter C in Eq. (2.21) need to be identified. Two independent test scenarios are defined to isolate each of these parameters and identify them one at a time.

3.1.1 Scenario 1: Straight motion

This scenario is an open-loop straight-line test. The different constant inputs U_1's each lasting 15 seconds are specified, while U_2's are kept zero at all times. This is achieved by giving constant propeller speed (n) and zero rudder angle (α). The boat is tested at three different speeds $n = 69.2$, 98, 120 rps. For each U_1 of the test the boat started from rest. This constitutes a straight-line motion with increasing speeds up to different terminal speeds for the corresponding n. For a straight-line motion, v and r and their corresponding derivatives are zero.

Identification of $\mathbf{d_{11}/A}$: With the linear motion of the boat, u speed component of the boat reaches a steady-state value. The derivative of u is equal to zero at this steady state. This constitutes a straight-line motion, with constant velocity of u_{ss}. These steady-state values are used for identification of d_{11}/A. Equation (3.1) reduces to

$$(d_{11}/A)u_{ss} = U_1, \tag{3.4}$$

$$ln(u_{ss}) + ln(d_{11}/A) = Cln(n) \quad where \quad U_1 = |n|^C, \tag{3.5}$$

$$ln(u_{ss}) = Cln(n) - ln(d_{11}/A) \tag{3.6}$$

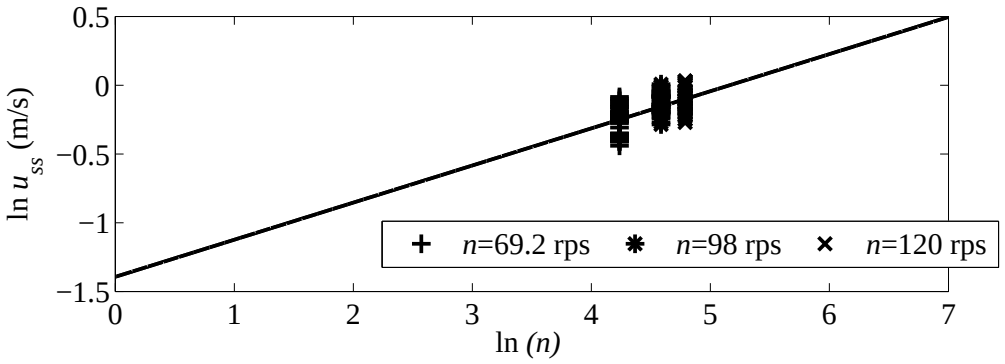


Figure 3.1: ln(n) Vs ln(u_{ss}); n in rps; u_{ss} in m/s

The Eq. (3.6) can be represented in $y = mx + K$ form, where $m = C$, $y = ln(u_{ss})$, $x = ln(n)$ and $K = -ln(d_{11}/A)$. The values of u_{ss} are recorded during each test at each constant speed n. A line is fitted to the pairs ($ln(n)$, $ln(u_{ss})$) (Figure 3.1)

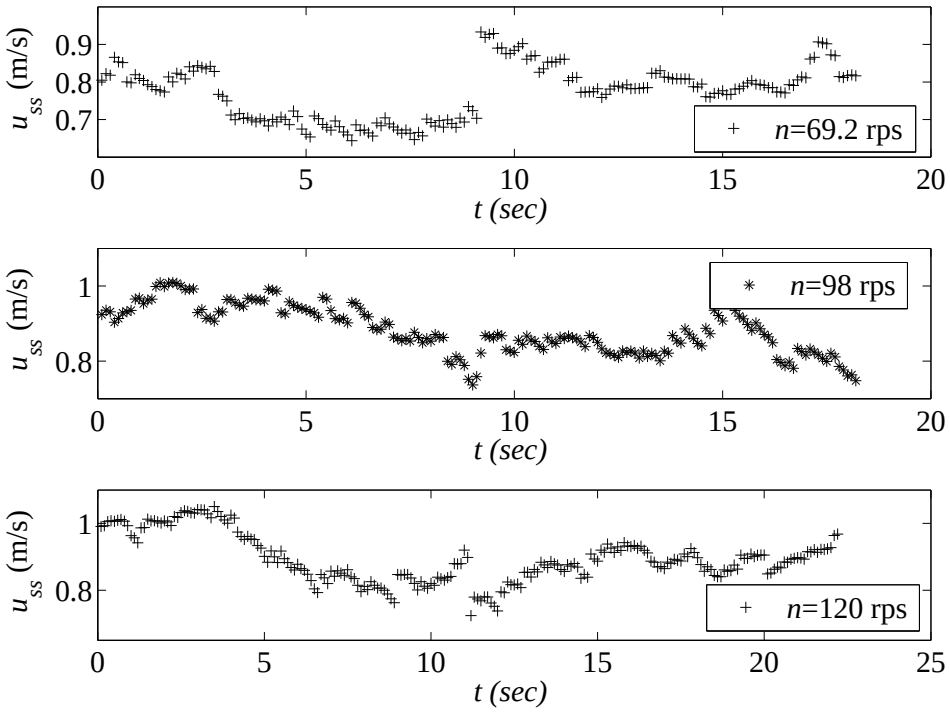


Figure 3.2: The variance in steady state longitudinal velocity of the boat

using the Least Square Method.

$$E = \sum_{i=1}^{N} \tfrac{1}{2}(y_i - y)^2, \tag{3.7}$$

$$\tfrac{\partial E}{\partial m} = \sum_{i=1}^{N}(mx_i + K - ln(u_{ss}))x_i = 0, \tag{3.8}$$

$$\tfrac{\partial E}{\partial K} = \sum_{i=1}^{N}(mx_i + K - ln(u_{ss})) = 0. \tag{3.9}$$

An error function E Eq. (3.7) is defined. The minimization of error is done by solving equations (3.8) and (3.9) simultaneously with the substitution of N data points from the three tests. Physically, the thrust of the vessel reduces nonlinearly as n increases. So, a $C < 1$ is expected and also the d_{11}/A should be positive. From the data sheet of the sensor in appendix Figure 8.2, accuracy of velocity in x and y

directions are 0.4 m/s (rms) and this caused u values in the Figure 3.2 to randomly distribute around the middle of actual u. The variance in each case is not more than 0.25 m/s which is well bellow the accuracy of the sensor. From the Figure 3.1, a line is fitted which passes through the middle of different u's and intercept K = -1.3937 and slope $m = 0.27$. The parameters $d_{11}/A = 4.0297$ and $C = 0.27$ are calculated.

Identification of $\mathbf{m_{11}/A}$: For determining m_{11}/A, the values of transient response of boat longitudinal velocity u are used. For a straight-line motion, v and r and corresponding derivative terms are zero. Eq. (3.1) reduces to

$$(m_{11}/A)\dot{u} + (d_{11}/A)u = U_1. \tag{3.10}$$

Solution to Eq. (3.10) is given by

$$u(t) = ce^{-t/\lambda} + U_1/(d_{11}/A), \quad \lambda = \tfrac{m_{11}}{A}/\tfrac{d_{11}}{A}, \tag{3.11}$$

where c is calculated by substituting initial velocity of the boat $u(0)$ in Eq. (3.11), i.e $c = u(0) - U_1/(d_{11}/A)$ and d_{11}/A is determined from previous test.

An error function E (Eq. (3.12)) is defined. The minimization of error function is done by equating partial derivative with respect to λ to zero Eq. (3.14). This gives

the values of λ at minimum error.

$$E = \sum_{i=1}^{N} \tfrac{1}{2}(u(t_i) - u_i)^2, \tag{3.12}$$

$$E = \sum_{i=1}^{N} \tfrac{1}{2}(ce^{-t_i/\lambda} + U_1/(d_{11}/A) - u_i)^2, \tag{3.13}$$

$$\tfrac{\partial E}{\partial \lambda} = \sum_{i=1}^{N} (ce^{-t_i/\lambda} + U_1/(d_{11}/A) - u_i)(t_i ce^{-t_i/\lambda})/\lambda = 0. \tag{3.14}$$

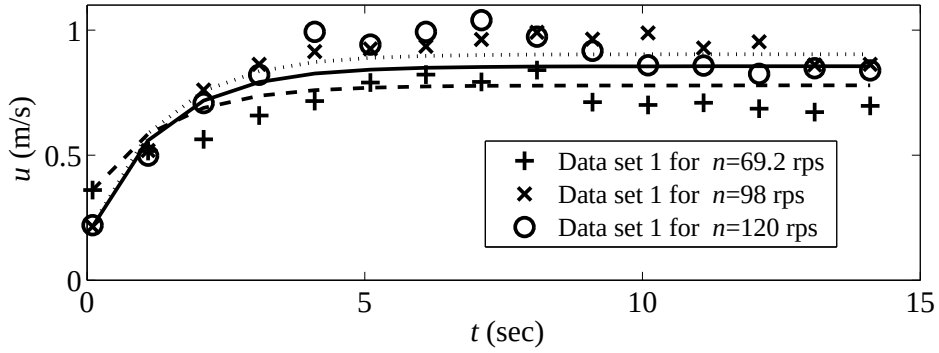


Figure 3.3: Curves fitted to data set 1 for identifying m_{11}/A

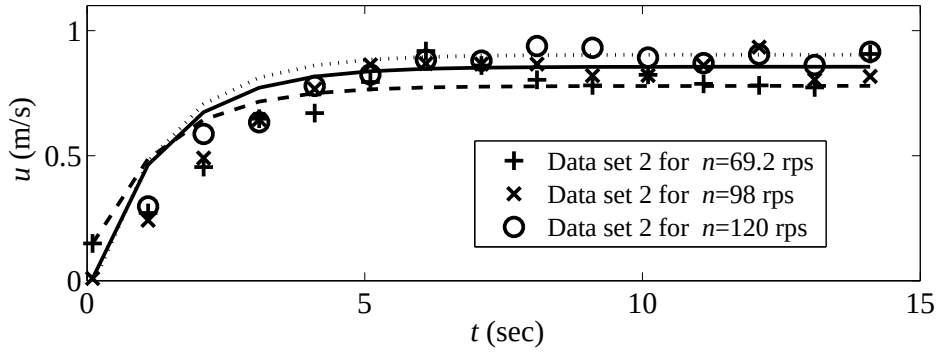


Figure 3.4: Curves plotted using identified m_{11}/A to data set 2 for verification

Each of three tests are performed twice, with two sets of data point for each test. One set of u's data points are substituted in Eq. (3.12). The respective plots

are fitted using Least Square Method (Figure 3.3) and $\lambda = 1.2985$ and $m_{11}/A = 5.2324$ are calculated. With the identified m_{11}/A, curves are plotted (Figure 3.4) and overlapped on top of the other set of data points. This plot is verification of the identified m_{11}/A.

3.1.2 Scenario 2: Circular motion

This scenario is an open-loop circular test. For these tests, boat is tested with constant propeller speed ($n = 98$ rps) and with five different rudder angles for five tests ($\alpha = 16°, 24°, -8°, -16°, -24°$) each lasting 20 sec. This generates different constant inputs U_1's and U_2's. For each test the boat starts nearly from rest. These constitutes a curved motions of the boat with increasing speeds to a constant circular motion with different terminal speeds. For the boat to generate a transient response, the rudder angle is set to zero for the first 5 sec and the rudder angle changes and remains constant for the rest of the test. With the circular motion of the boat u, v and r speed component of the boat reach a steady-state value. The terms $\dot{u}$, $\dot{v}$ and $\dot{r}$ are equal to zero at this steady state. So, the equations (3.1), (3.2) and (3.3) reduce to

$$(m_{22}/A)vr = (d_{11}/A)u - U_1, \tag{3.15}$$

$$(d_{22}/A)v = -(m_{11}/A)ur + U_2, \tag{3.16}$$

$$(d_{66}/A)r = ((m_{11} - m_{22})/A)uv - LU_2. \tag{3.17}$$

The equations (3.15), (3.16) and (3.17) can be represented in $y = mx + K$ form. The steady-state values of u_{ss}, v_{ss}, r_{ss}, and n, α collected from recorded data are substituted in these equations

$$(d_{11}/A)u_{ss} = (m_{22}/A)v_{ss}r_{ss} + |n|^C \cos\alpha, \tag{3.18}$$

$$(m_{11}/A)u_{ss}r_{ss} = -(d_{22}/A)v_{ss} + |n|^C \sin\alpha, \tag{3.19}$$

$$((m_{11} - m_{22})/A)u_{ss}v_{ss} = (d_{66}/A)r_{ss} + L|n|^C \sin\alpha. \tag{3.20}$$

$$E = \sum_{i=1}^{N} \tfrac{1}{2}(mx_i + K - y_i)^2,$$
$$\tfrac{\partial E}{\partial m} = \sum_{i=1}^{N}(mx_i + K - y_i)x_i = 0. \tag{3.21}$$

The error function E is defined. The minimization of error function is done by equating Eq. (3.21) to zero which gives the values of m at minimum error.

Identification of $\mathbf{m_{22}/A}$: Using Least Square Method, a line to the steady state data points is fitted using Eq. (3.18). From five tests performed, two tests ($\alpha = -24^\circ, 24^\circ$) data points are substituted in Eq. (3.21) and a line with slope $m = m_{22}/A = 5.9302$ is calculated (Figure 3.5). With the identified m_{22}/A, a line is plotted (Figure 3.6) and overlapped on top of the other three set of tests data points. This plot is verification of the identified m_{22}/A.

Identification of $\mathbf{d_{22}/A}$: Using Least Square Method, a line to the steady state data points is fitted using Eq. (3.19). From five tests performed, two tests

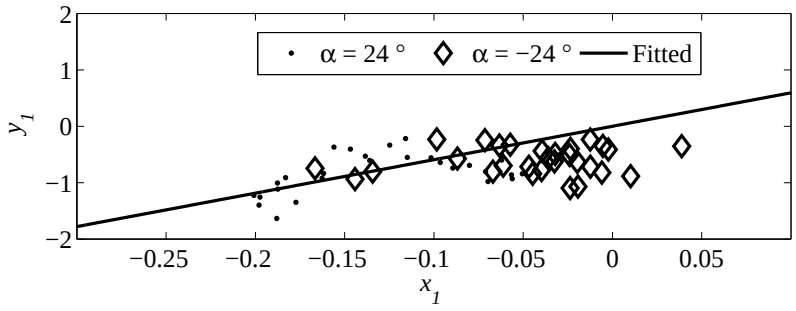


Figure 3.5: Line fitted to data points of rudder angle 24° and −24° for identifying m_{22}/A. From Eq. (2.18) $x_1 = v_{ss}r_{ss}$, $y_1 = (d_{11}/A)u_{ss}$

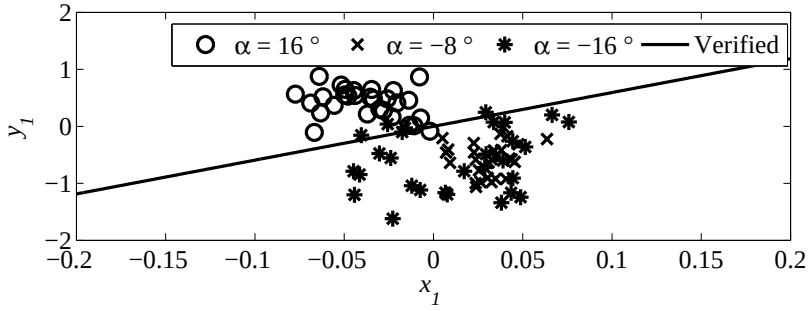


Figure 3.6: Line plotted using identified m_{22}/A on data points of rudder angle 16°, −8° and −16° for verification. From Eq. (2.18) $x_1 = v_{ss}r_{ss}$, $y_1 = (d_{11}/A)u_{ss}$

($\alpha = -24°, 24°$) data points are substituted in Eq. (3.21) and a line with slope $m = d_{22}/A = 5.4845$ is calculated (Figure 3.7). With the identified d_{22}/A, a line is plotted (Figure 3.8) and overlapped on top of the other three set of tests data points. This plot is verification of the identified d_{22}/A.

Identification of $\mathbf{d_{66}/A}$: Using Least Square Method, a line to the steady state data points is fitted using Eq. (3.20). From five tests performed, two tests ($\alpha = -24°, 24°$) data points are substituted in Eq. (3.21) and a line with slope $m = d_{66}/A = 1.1276$ is calculated (Figure 3.9). With the identified d_{66}/A, a line is plotted

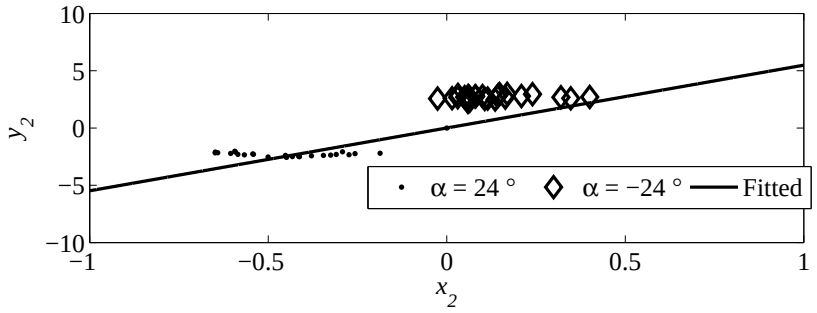


Figure 3.7: Line fitted to data points of rudder angle 24° and −24° for identifying d_{22}/A. From Eq. (2.19) $x_2 = -v_{ss}$, $y_2 = (m_{11}/A)u_{ss}r_{22}$

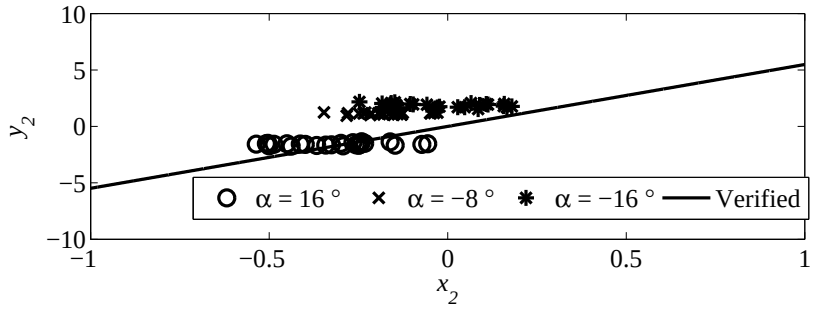


Figure 3.8: Line plotted using identified d_{22}/A on data points of rudder angle 16°, −8° and −16° for verification. From Eq. (2.19) $x_2 = -v_{ss}$, $y_2 = (m_{11}/A)u_{ss}r_{22}$

(Figure 3.10) and overlapped on top of the other three set of tests data points. This plot is verification of the identified d_{66}/A.

Identification of $\mathbf{I_{33}/A}$: The transient response of yaw rate (r) of the boat is used for the identification of I_{33}/A.

$$(I_{33}/A)\dot{r} + (d_{66}/A)r = ((m_{22} - m_{11})/A)uv - LU_2. \tag{3.22}$$

As the sensor does not output $\dot{r}$, Eq. (3.22) can not be linearly solved. The Eq. (3.22) is a linear first order ODE with the values of u and v varying. So finding

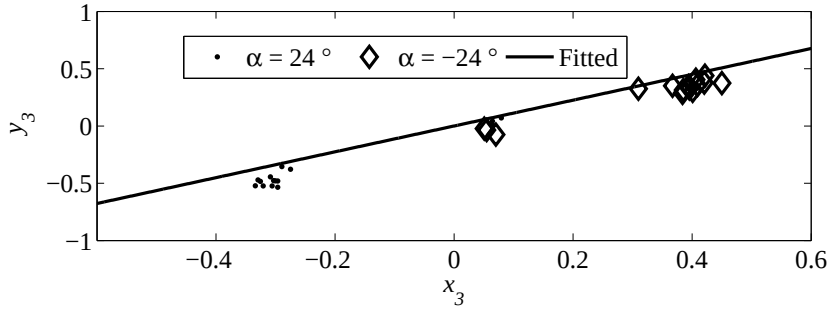


Figure 3.9: Line fitted to data points of rudder angle 24° and −24° for identifying d_{66}/A. From Eq. (2.20) $x_3 = r_{ss}$, $y_3 = ((m_{11} - m_{22})/A)u_{ss}v_{22}$

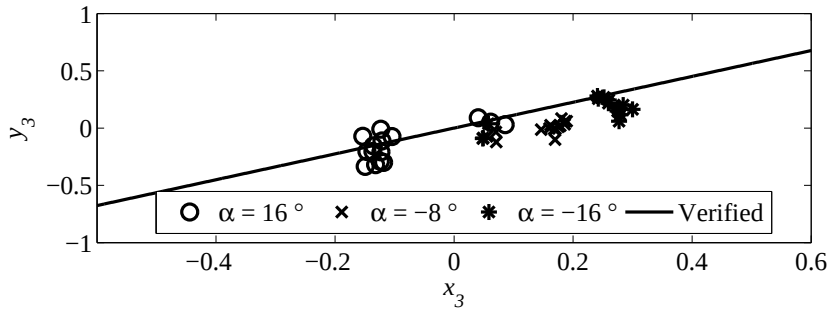


Figure 3.10: Line plotted using identified d_{66}/A on data points of rudder angle 16°, −8° and −16° for verification. From Eq. (2.20) $x_3 = r_{ss}$, $y_3 = ((m_{11} - m_{22})/A)u_{ss}v_{22}$

a closed form solution and solving for I_{33}/A using least square is very difficult. For the simplified closed form solution using Least Square Method, exponential curves are fitted to u and v data points.

$$u = Ae^{B(t-5)} + C \tag{3.23}$$

$$v = De^{E(t-5)} + F \tag{3.24}$$

The error function E_1 and E_2 are defined.

$$E_1 = \sum_{i=1}^{N} \tfrac{1}{2}(Ae^{B(t(i)-5)} + C - u_i)^2, \tag{3.25}$$

$$E_2 = \sum_{i=1}^{N} \tfrac{1}{2}(De^{E(t(i)-5)} + F - v_i)^2, \tag{3.26}$$

$$\tfrac{\partial E_1}{\partial A} = \sum_{i=1}^{N}(Ae^{B(t(i)-5)} + C - u_i)e^{B(t(i)-5)} = 0, \tag{3.27}$$

$$\tfrac{\partial E_1}{\partial B} = \sum_{i=1}^{N}(Ae^{B(t(i)-5)} + C - u_i)A(t(i)-5)e^{B(t(i)-5)} = 0, \tag{3.28}$$

$$\tfrac{\partial E_1}{\partial C} = \sum_{i=1}^{N}(Ae^{B(t(i)-5)} + C - u_i) = 0, \tag{3.29}$$

$$\tfrac{\partial E_2}{\partial D} = \sum_{i=1}^{N}(De^{E(t(i)-5)} + F - v_i)e^{E(t(i)-5)} = 0, \tag{3.30}$$

$$\tfrac{\partial E_2}{\partial E} = \sum_{i=1}^{N}(De^{E(t(i)-5)} + F - v_i)D(t(i)-5)e^{E(t(i)-5)} = 0, \tag{3.31}$$

$$\tfrac{\partial E_2}{\partial F} = \sum_{i=1}^{N}(De^{E(t(i)-5)} + F - v_i) = 0. \tag{3.32}$$

For the first 5 sec, boat moves in straight path, so the transient response of yaw rate of boat from 5 to 10 sec is used. The data points for the tests of rudder angle (α) 24° and $-24°$ are used. Each test has two sets of data points. The two sets of data points for u and v are substituted in Eq.s (3.27) to (3.32) and corresponding values in Table 3.1 are calculated. The Figure 3.11 and Figure 3.12 shows the fitted curves for $\alpha = 24°$ and $-24°$.

Substituting u and v in Eq. (3.22)

$$\lambda\dot{r} + r = K(Ae^{B(t-5)} + C)(De^{E(t-5)} + F) - LU_2/(d_{66}/A). \tag{3.33}$$

Table 3.1: Fitting u and v curve parameters

Parameters	$\alpha = 24°$	α = -24°
A	0.1651	-0.0151
B	-1.0254	0.4646
C	0.5656	0.7170
D	-0.2062	0.4717
E	-1.4057	-0.3538
F	0.2167	-0.2356

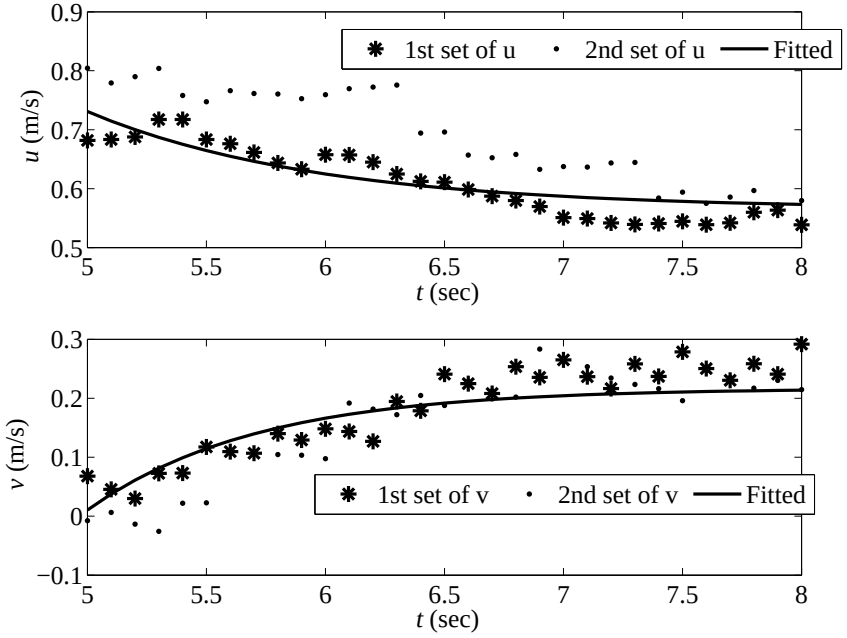


Figure 3.11: Fitted curves for transient response of u and v for $\alpha = 24°$

where $\lambda = \frac{I_{33}/A}{d_{66}/A}$ and $K = \frac{(m_{22}-m_{11})/A}{d_{66}/A}$.

The closed form solution for r is given by

$$r(t) = c1e^{\frac{-t}{\lambda}} - LU_2/(d_{66}/A) + \tag{3.34}$$

$$\left(\frac{AFe^{B(t-5)}}{1+\lambda B} + \frac{ADe^{(B+E)(t-5)}}{1+\lambda B+\lambda E} + C\left(F + \frac{De^{E(t-5)}}{1+\lambda E}\right)\right)K. \tag{3.35}$$

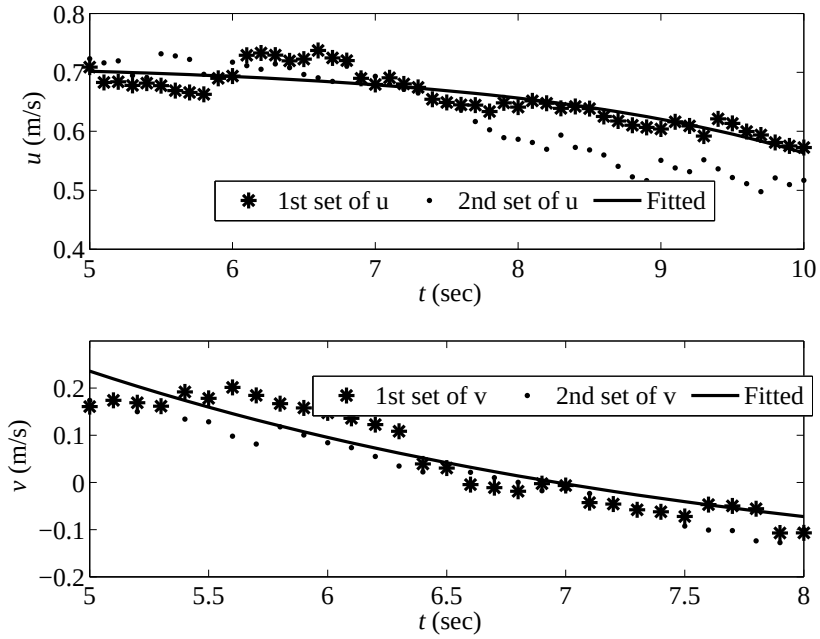


Figure 3.12: Fitted curves for transient response of u and v for $\alpha = -24^\circ$

$$E = \sum_{i=1}^{N} \frac{1}{2}(r(t_i) - r_i)^2, \tag{3.36}$$

$$\frac{\partial E}{\partial \lambda} = \sum_{i=1}^{N} (r(t_i) - r_i)\frac{\partial r(t_i)}{\partial \lambda}. \tag{3.37}$$

The data points for rudder angles 24° and -24° are considered. The two sets of data points for $\alpha = 24^\circ$ are substituted in Eq. (3.37). Substituting A, B, C, D, E and F for $\alpha = 24^\circ$ and previously determined parameters using Least Square Method respective curves are fitted(Figure 3.13). The values of $\lambda = 1.1057$ and corresponding $I_{33}/A = 1.24678732$ are identified. With the identified I_{33}/A, substituting A, B, C,

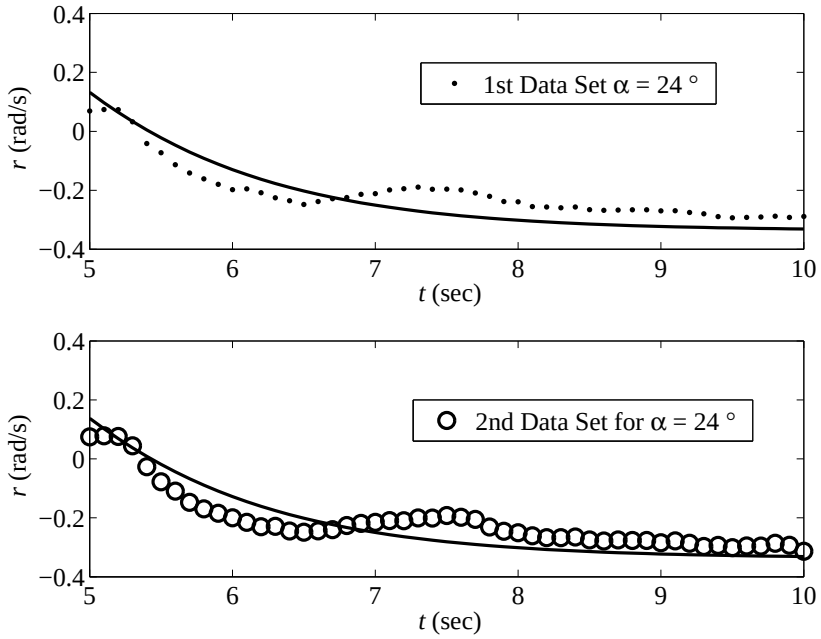


Figure 3.13: Curves fitted to the two data sets of $\alpha = 24°$ for identifying I_{33}/A

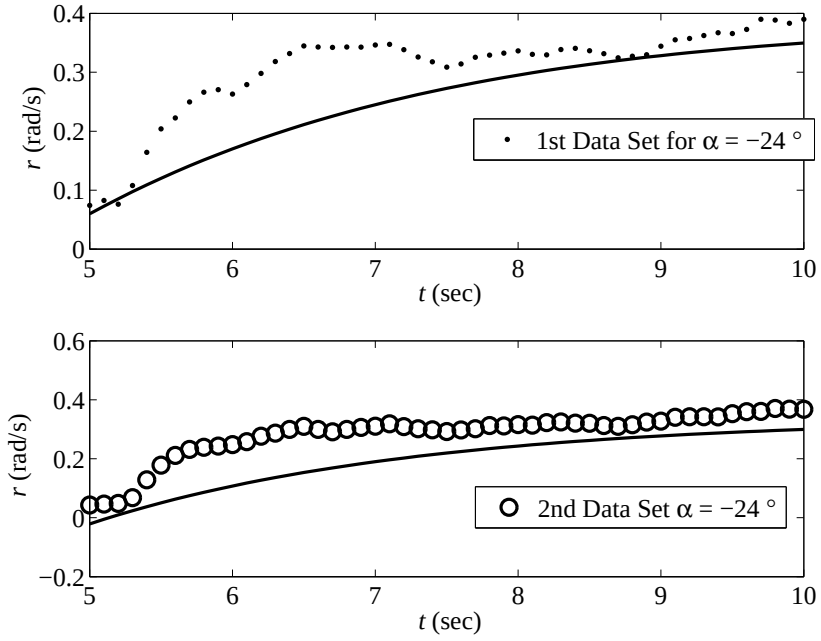


Figure 3.14: Curves plotted using identified I_{33}/A to the two data sets of $\alpha = -24°$ for verification

D, E and F for $\alpha = -24°$, curves are plotted (Figure 3.14) and overlapped on top of two sets of data points. This plot is verification of identified I_{33}/A.

3.2 Simulation of model with the identified parameters:

The identified parameters are substituted in Eqs (2.8) and the inputs n and α are applied to find the simulated response of the state variables.

3.2.1 Straight motion:

The simulated and tested response for the straight line test ($n = 98$ rps and $\alpha = 0$) are plotted. As it being not a closed loop test the simulated and tested responses does not match exactly. The Figure 3.15 shows the x, y and θ response, at the start of the test for 3 seconds, due to the disturbance the error gets added up in time and deviates from simulated response. The disturbance in lateral motion and rotation of the boat can be seen in Figure 3.16 and the error in these values also very less when compared with large u values. The simulated v and r values goes to zero s it being a straight line test. Following a straight path accurately under disturbance is not possible but when compared with the simulations the boat is following the simulated path closely.

3.2.2 Circular motion:

The simulated and tested response for the circular test ($n = 98$ rps and $\alpha = 0$ for 5 seconds and -24° for the rest of the time) are plotted. The Figure 3.17 shows the x, y and θ values follow the simulated values for first 5 sec as it being a straight

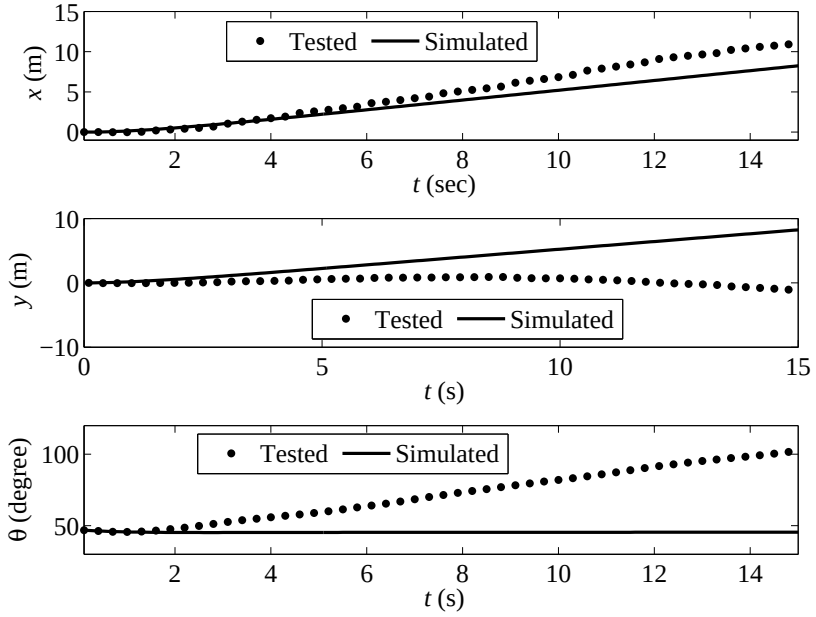


Figure 3.15: Simulated and tested x, y and θ for straight motion of the boat

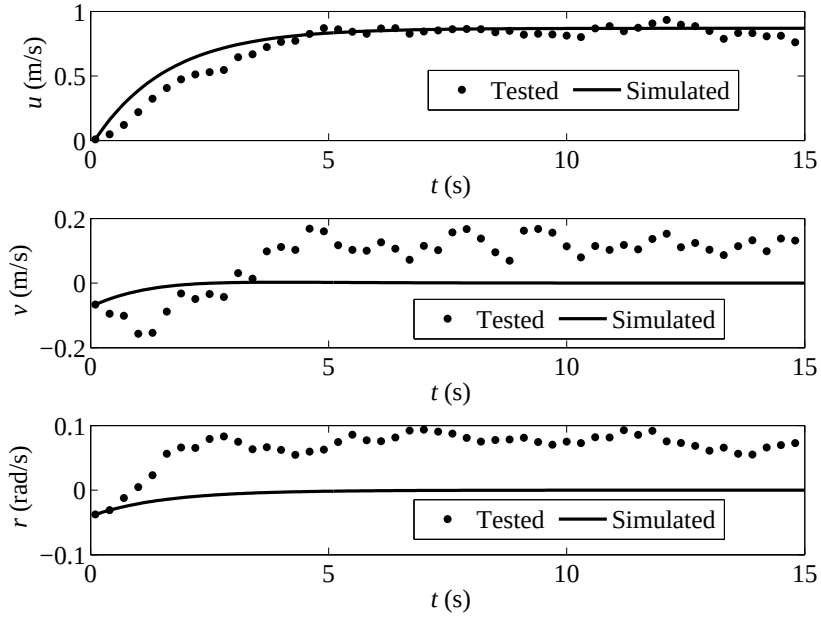


Figure 3.16: Simulated and tested u, v and r velocities for straight motion of the boat

line and as α changes suddenly these deviate from simulated values but the shape of these curves are same with an offset error. The Figure 3.18 show the velocities of boat in all 3 DOFs. The u value rises to steady state value and after 5 seconds it goes down because of additional lateral component of the boat. The sudden change in α causes the rotational velocity r to rise. All these values reach a steady state value but the difference in simulated and tested v steady state values shows clearly the underactuated nature of the boat. The Figure 3.19 shows the path of the boat being straight for 5 sec and circular after that. The shape of the curves are same but as disturbance grows in time these deviate away. From these plots the accuracy of the identified model parameters is verified.

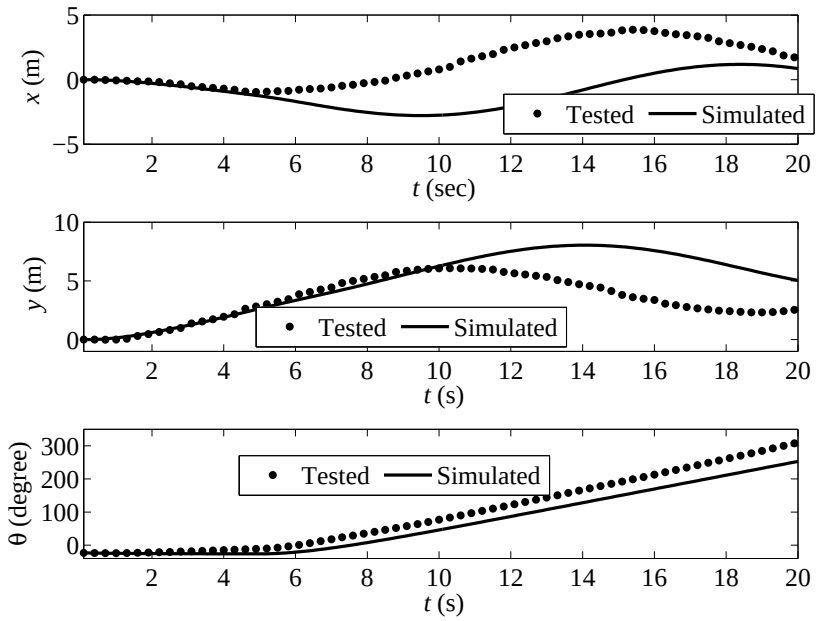


Figure 3.17: Simulated and tested x, y and θ for circular motion of the boat

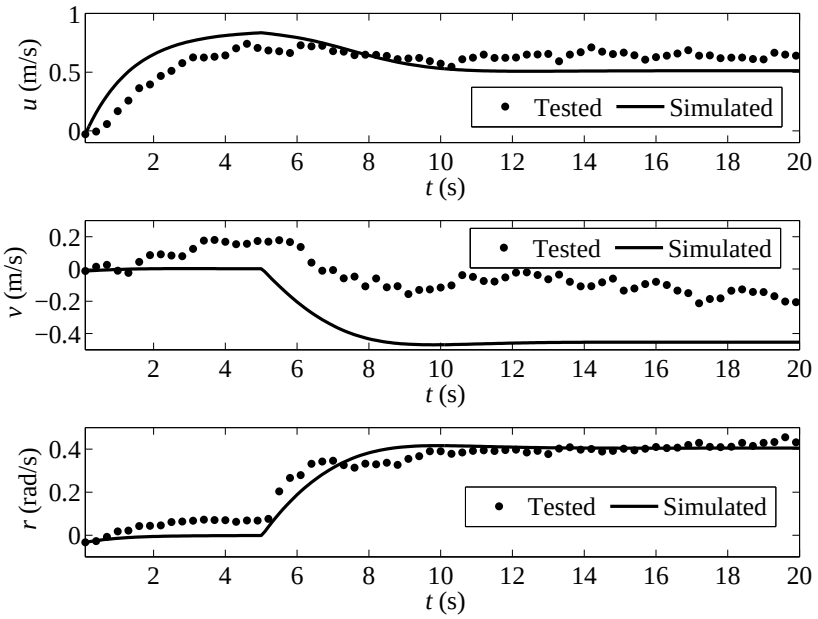


Figure 3.18: Simulated and tested u, v and r velocities for circular motion of the boat

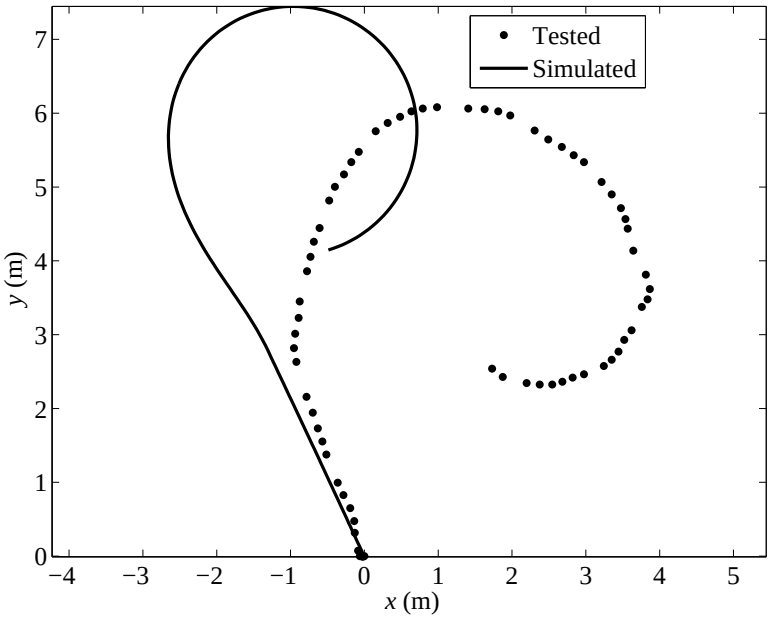


Figure 3.19: The tested and simulated for circular path of the boat

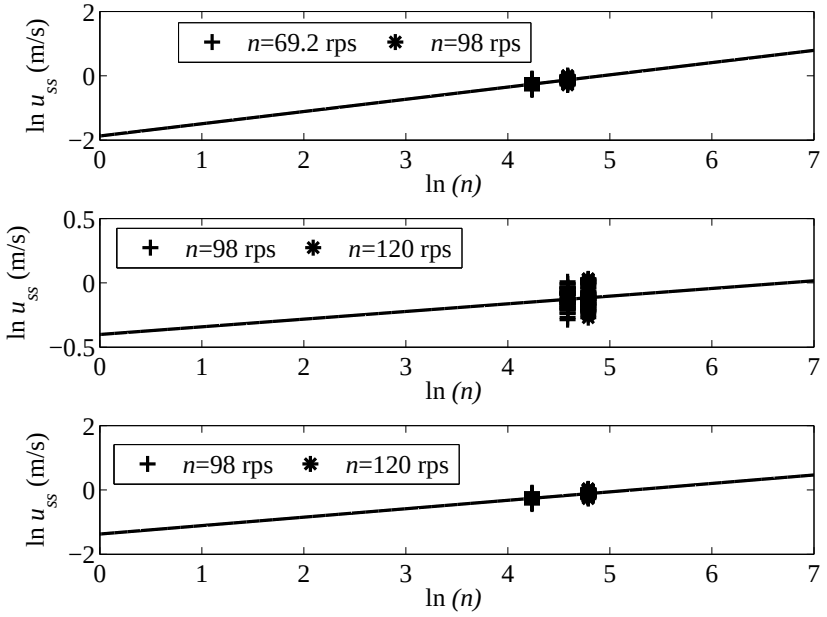


Figure 3.20: The fitted lines to all the three combinations of data points for determining $d11/A$ and C

3.3 Analysing the sensitivity of parameters

From the Figure 3.1, the data points used are randomly distributed because of sensor, this affects d_{11}/A and C values. The parameters determined based on these values could be different and changes the dynamics. For checking the sensitivity of these parameters with the change of data points, using two combination of three data points sets i.e $n = 69.2$, 98 rps, $n = 98$, 120 rps and $n = 69.2$, 120 rps. The Figure 3.20 shows the fitted plots for determining d_{11}/A and C all these combinations. The values in the order are $d_{11}/A = 6.4971$, 1.4937, 3.9415, $C = 0.3803$, 0.0596, 0.2623 respectively. Using these values the other parameters are determined based on the method used before. The values are given in the Table 3.2.

Table 3.2: The identified parameters using all combinations of data points

Parameter	Data points 1,2,3	Data points 1,2	Data points 2,3	Data points 1,3
m_{11}/A	5.2324	9.4344	1.9664	4.9348
m_{22}/A	5.9302	10.6415	2.4438	5.4989
I_{33}/A	1.24678732	2.0936	0.5887	1.1479
d_{11}/A	4.0297	6.4971	1.4937	3.9415
d_{22}/A	5.4845	9.4247	2.0781	5.2436
d_{66}/A	1.1276	1.8921	0.524	1.0397
C	0.27	0.3803	0.0596	0.2623

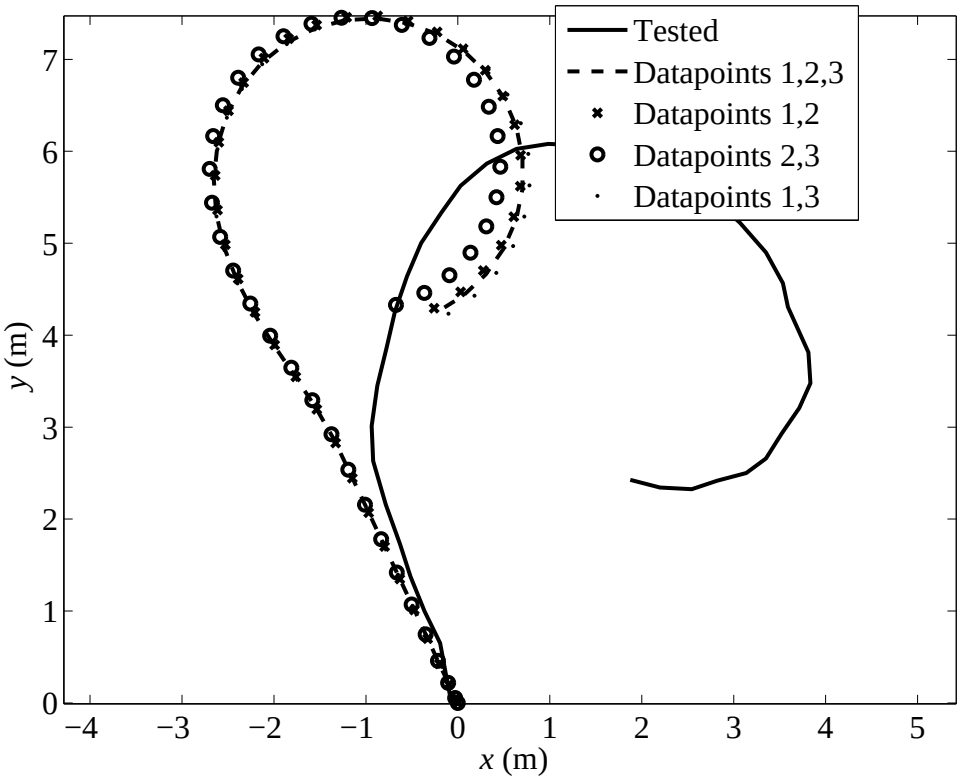


Figure 3.21: The simulated response of the boat for different combinations of parameters determined on the actual motion

The Figure 3.21 shows using all the above values, the simulated dynamic motion of the boat on top of actual motion to check how the different sets of these parameters effects the dynamics. It can be deduced from Figure 3.21 that all the

simulated paths for these four sets of parameters are similar and thus little effect on the dynamics of the boat.

3.4 Summary:

The tests are conducted in a stagnant water with less disturbance on the boat for making dynamic model more accurate. Each test is performed twice to check the consistency of the tests. Table 6.1 shows identified parameters through various tests. These identified values are used for simulation of the boat controller (Nonlinear Model Predictive Control) and in design of embedded control program in Simulink.

Table 3.3: Numerical results for the identified parameters

Parameter	Identification result	Units
m_{11}/A	5.2324	Kg/unit A
m_{22}/A	5.9302	Kg/unit A
I_{33}/A	1.24678732	Kg m^2/unit A
d_{11}/A	4.0297	N s m^{-1}/unit A
d_{22}/A	5.4845	N s m^{-1}/unit A
d_{66}/A	1.1276	N m s rad^{-1}/unit A

Note: Unit A is N rps^{-C}, where $C = 0.27$.

CHAPTER 4

CONTROLLER DESIGN

The control inputs for the boat are propeller speed (n) and rudder angle (α) which generate the required U_1 and U_2 for tracking the desired path. The controller calculates these values based on the error in state vector from desired. There is no guarantee that the servo motors on the boat can generate these calculated values. If the desired trajectory is not defined with care such that a sudden change in velocity or a shape change in path, calculates larger propeller speed or larger rudder angle which is too high for the servo motors to generate. This causes the saturation of the inputs. Even if the inputs are constrained using a simple if command, it leads to dynamic instability of boat and loosing the track of the path. In this case, a better controller which limits the control inputs to a maximum value and anticipates the sudden change in inputs and recovers to its path after saturation has to be designed. Model Predictive control, which determines a series of future control commands based on a real-time optimization, is a good method for controlling systems with constraints.

4.1 Nonlinear Model Predictive Control (NMPC) Formulation

In the NMPC method, at any time 't' during real time control, the future control inputs ($\mathbf{U}$) are predicted for T sec into the future. The length of T which spans multiple sampling periods, is called Prediction horizon. The controller applies only the T_1 part of T inputs, as the predicted values deviate from actual values as time passes. This T_1 is called Control horizon. Normally T_1 equal to the sampling period. Based on the state variables at time t, a cost function is optimized to determine the future control action.

$$\mathbf{U}(t) \quad \tau \, \epsilon \, [t, t+T], \tag{4.1}$$

such that the cost function

$$J = \phi(\mathbf{z}(t+T)) + \int_{t}^{t+T} L(\mathbf{z}(\tau), \mathbf{U}(\tau)) \, \mathrm{d}\tau, \tag{4.2}$$

where $\mathbf{z} = [\ x_p,\ y_p]^T$ contains control outputs of the system.

Solving analytically for $\mathbf{U}(\tau)$ is very difficult for nonlinear model systems. Therefore numerical methods are used. The future control action for time T is assumed to be a stepwise function with N steps. Each step width of $\Delta t = T/N$ and height of the step represents control input for Δt sec.

Discretized cost function and control inputs are given by

$$\mathbf{U}_k = \quad \mathbf{U}(t + k\Delta t), \quad k = 0, ...N - 1 \tag{4.3}$$

$$J = \quad \phi(\mathbf{z}_N) + \sum_{k=0}^{N-1} L_k(\mathbf{z}_k, \mathbf{U}_k)\Delta t. \tag{4.4}$$

Constraint equations which prevent input saturation must be defined

$$U_i \leq U_{i\,\max} \quad i = 1, .., m \tag{4.5}$$

where m is the number of inputs. The inequalities in the U_i's must be formulated to equalities to use them in optimization function. To convert inequalities to equalities, dummy control inputs U_j's ($j = m + 1, ..., 2m$) are defined such that the equalities can be written in the general form of constraint equations

$$\mathbf{C}(\mathbf{U}) = \mathbf{0}. \tag{4.6}$$

where $\mathbf{U}$ is a new augmented control input vector that consists of both actual and dummy control inputs. The number of dummy inputs are equal to number of control inputs, and these do not effect the dynamics of the vessel but it conveys, the amount of offset of the control inputs from the maximum values to the controller.

$$\mathbf{U} = \begin{bmatrix} U_1 & U_2 & ... & U_m & U_{m+1} & ... & U_{2m} \end{bmatrix}^T. \tag{4.7}$$

The state space representation of dynamic model of the system (Eq. (2.26)) is given by

$$\dot{\mathbf{q}} = \mathbf{f}(\mathbf{q}, \mathbf{U}). \tag{4.8}$$

The discretized dynamic model is

$$\begin{aligned} \mathbf{q}_{k+1} = & \quad \mathbf{q}_k + \mathbf{f}(\mathbf{q}_k, \mathbf{U}_k)\Delta t \\ = & \quad \mathbf{f}_k \end{aligned} \tag{4.9}$$

The future discretized control inputs $\mathbf{U}_k$, $k = 0, ...N-1$ at any control time t are calculated by minimized the discretized cost function Eq. (4.2) in real-time. The calculated U's from minimization problem must be compatible with the dynamics of the system and must satisfy the constraints. So the cost function is augmented with zero terms of dynamic model Eq. (4.9) and constraints Eq. (4.6).

$$J = \phi(\mathbf{z}_N) + \sum_{k=0}^{N-1} [L_k + \boldsymbol{\lambda}_{k+1}^T(\mathbf{f}_k - \mathbf{q}_{k+1}) + \boldsymbol{\mu}_k^T \mathbf{C}_k]\Delta t, \tag{4.10}$$

where $\boldsymbol{\lambda}_k$'s are the costates and $\boldsymbol{\mu}_k$'s are the Lagrange multipliers.

$$\phi(\mathbf{z}_N) = \frac{1}{2}(\mathbf{z}_N - \mathbf{z}_N^d)^T \mathbf{Z}_0 (\mathbf{z}_N - \mathbf{z}_N^d), \tag{4.11}$$

$$\begin{aligned} L_k = & \frac{1}{2}((\mathbf{z}_k - \mathbf{z}_k^d)^T \mathbf{Z}(\mathbf{z}_k - \mathbf{z}_k^d) + (\mathbf{U}_k - \mathbf{U}_k^d)^T \mathbf{R}(\mathbf{U}_k - \mathbf{U}_k^d) \\ & -\mathbf{S}(\mathbf{U}_k - \mathbf{U}_k^d)). \end{aligned} \tag{4.12}$$

where $\mathbf{Z_0}$ and $\mathbf{Z}$ are 2×2 positive definite diagonal weight matrices which penalize the control outputs errors. $\mathbf{R}$ is a 4×4 positive semi definite diagonal weight matrices penalizing the real control actions. As U_3 and U_4 are dummy inputs, these should not be penalized. However, when U_3 and U_4 equal to zero the solution of the cost function can bifurcate. So a 4×4 positive semi definite dummy penalty weight matrix $\mathbf{S}$ is added and it should not penalize the U_1 and U_2.

$$\mathbf{R} = \begin{bmatrix} R_{11} & 0 & 0 & 0 \\ 0 & R_{22} & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad \mathbf{S} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & S_{33} & 0 \\ 0 & 0 & 0 & S_{44} \end{bmatrix}. \tag{4.13}$$

For formulating the cost function in the simple form. Let

$$H_k = L_k + \boldsymbol{\lambda}_{k+1}^T \mathbf{f}_k + \boldsymbol{\mu}_k^T \mathbf{C}_k, \tag{4.14}$$

$$J = \phi(\mathbf{z}_N) + \sum_{k=0}^{N-1} (H_k - \boldsymbol{\lambda}_{k+1}^T \mathbf{q}_{k+1}) \Delta t. \tag{4.15}$$

Rearranging the terms and indices in the summation of Eq. (4.15)

$$J = \phi(\mathbf{z}_N) + \boldsymbol{\lambda}_N^T \mathbf{q}_N \Delta t + H_0 \Delta t + \sum_{k=1}^{N-1} (H_k - \boldsymbol{\lambda}_k^T \mathbf{q}_k) \Delta t. \tag{4.16}$$

Differential of cost function J is given by

$$dJ = \sum_{k=0}^{N} \frac{\partial J}{\partial \mathbf{q}_k^T} d\mathbf{q}_k + \sum_{k=0}^{N-1} \left(\frac{\partial J}{\partial \mathbf{U}_k^T} d\mathbf{U}_k + \frac{\partial J}{\partial \boldsymbol{\mu}_k^T} d\boldsymbol{\mu}_k \right). \tag{4.17}$$

The minimization of cost function is done by equating dJ to zero. This is done by equating all partial derivatives to zero which gives a system of equations, solving for $\mathbf{U_k}$ gives the discretized future control inputs. Here the control output $\mathbf{z}$'s and state space vector $\mathbf{q}$'s are different, for some systems these can be same. Only global position of control point is controlled. The orientation and velocities are not controlled as these values dependent on control outputs they are indirectly controlled.

$$\frac{\partial J}{\partial \mathbf{q}_k^T} = \frac{\partial J}{\partial \mathbf{z}_k^T} \frac{\partial \mathbf{z_k}}{\partial \mathbf{q}_k^T}. \tag{4.18}$$

The partial derivative of J with respect to states is

$$\frac{\partial J}{\partial \mathbf{q}_0^T} d\mathbf{q}_0 = \frac{\partial H_0}{\partial \mathbf{q}_0^T} \Delta t d\mathbf{q}_0, \tag{4.19}$$

$$\frac{\partial J}{\partial \mathbf{q}_k^T} d\mathbf{q}_k = (\frac{\partial H_k}{\partial \mathbf{q}_k^T} - \boldsymbol{\lambda}_k^T) \Delta t d\mathbf{q}_k, \quad k = 1, ..., N-1 \tag{4.20}$$

$$\frac{\partial J}{\partial \mathbf{q}_N^T} d\mathbf{q}_N = (\frac{\partial \phi}{\partial \mathbf{q}_N^T} - \boldsymbol{\lambda}_N^T \Delta t) d\mathbf{q}_N. \tag{4.21}$$

At initial states $\mathbf{q_0}$, the control input calculations are done and these states are not functions of future control actions, so d$\mathbf{q}_0$ is zero. The partial derivative of Eq. (4.19) is zero and similarly the partial derivatives of Eqs. (4.20) and (4.21) must be made zero, as costates $\boldsymbol{\lambda}_k$'s are arbitrary numbers and these are chosen appropriately such that corresponding partial derivative are equal to zero.

$$\begin{aligned}
\boldsymbol{\lambda}_N^T &= \frac{\partial \phi}{\partial \mathbf{q}_N^T}\frac{1}{\Delta t}, \\
&= \left(\frac{\partial \phi}{\partial \mathbf{z}_N^T}\ \frac{\partial \mathbf{z}_N}{\partial \mathbf{q}_N^T} \right)\frac{1}{\Delta t}, \qquad (4.22)
\end{aligned}$$

$$\begin{aligned}
\boldsymbol{\lambda}_k^T &= \frac{\partial H_k}{\partial \mathbf{q}_k^T} \\
&= \frac{\partial L_k}{\partial \mathbf{q}_k^T} + \boldsymbol{\lambda}_{k+1}^T \frac{\partial \mathbf{f}_k}{\partial \mathbf{q}_k^T} + \boldsymbol{\mu}_k^T \frac{\partial \mathbf{C}_k}{\partial \mathbf{q}_k^T}, \\
&= \frac{\partial L_k}{\partial \mathbf{z}_k^T}\ \frac{\partial \mathbf{z}_k}{\partial \mathbf{q}_k^T} + \boldsymbol{\lambda}_{k+1}^T \frac{\partial \mathbf{f}_k}{\partial \mathbf{q}_k^T} + \boldsymbol{\mu}_k^T \frac{\partial \mathbf{C}_k}{\partial \mathbf{q}_k^T}, \quad k = N-1, ..., 1. \qquad (4.23)
\end{aligned}$$

The partial derivative of J with respect to control inputs $\mathbf{U}_k$'s is

$$\begin{aligned}
\frac{\partial J}{\partial \mathbf{U}_k^T} &= \frac{\partial H_k}{\partial \mathbf{U}_k^T} = \mathbf{0}, \\
&= \frac{\partial L_k}{\partial \mathbf{U}_k^T} + \boldsymbol{\lambda}_{k+1}^T \frac{\partial \mathbf{f}_k}{\partial \mathbf{U}_k^T} + \boldsymbol{\mu}_k^T \frac{\partial \mathbf{C}_k}{\partial \mathbf{U}_k^T} = \mathbf{0}, \quad k = 0, ..., N-1. \qquad (4.24)
\end{aligned}$$

The partial derivative of J with respect to Lagrange multipliers $\boldsymbol{\mu}_k$'s is

$$\begin{aligned}
\frac{\partial J}{\partial \boldsymbol{\mu}_k^T} &= \frac{\partial H_k}{\partial \boldsymbol{\mu}_k^T} = \mathbf{0}, \\
&= \mathbf{C}_k = \mathbf{0}, \quad k = 0, ..., N-1. \qquad (4.25)
\end{aligned}$$

The costates are calculated by solving Eqs. (4.22) and (4.23). These are substituted in Eq.s (4.24) and (4.25) and solved simultaneously for future control command

steps $\mathbf{U}_k$ and Lagrange multipliers $\boldsymbol{\mu}_k$. The total unknowns are $4N$ components of $\mathbf{U}_k$'s and $2N$ components of $\boldsymbol{\mu}$'s, cumulating all these unknowns into a single $6N \times 1$ vector $\mathbf{u}$.

$$\mathbf{u} = [\mathbf{U}_0^T \boldsymbol{\mu}_0^T ... \mathbf{U}_k^T \boldsymbol{\mu}_k^T ... \mathbf{U}_{N-1}^T \boldsymbol{\mu}_{N-1}^T]. \tag{4.26}$$

Representing system of $6N$ nonlinear equations (4.24) and (4.25) in matrix in the form of $\mathbf{U}$ vector.

$$\mathbf{F}(\mathbf{u}, \mathbf{q}_0, t) = \begin{bmatrix} (\partial H_0 / \partial \mathbf{U}_0^T)^T \\ \mathbf{C}_0 \\ \vdots \\ \vdots \\ (\partial H_k / \partial \mathbf{U}_k^T)^T \\ \mathbf{C}_k \\ \vdots \\ \vdots \\ (\partial H_{N-1} / \partial \mathbf{U}_{N-1}^T)^T \\ \mathbf{C}_{N-1} \end{bmatrix} = \mathbf{0}. \tag{4.27}$$

4.2 Applying to The Dynamic Model of The Vessel

Above derived formulations are applied to the dynamic model of the vessel. The physical limits of boat propeller and rudder are used in developing the input constraint equations. The control inputs U_1 and U_2 should be formulated into an equation such that these stay within the saturation limit. The maximum speed of the

propeller is n_{max} and maximum rudder angle is α_{max}. The maximum thrust generated by the propeller is given by

$$F_{max} = A|n_{max}|^C. \tag{4.28}$$

The resultant propeller thrust acting on the vessel should be less than or equal to F_{max}. The first inequality constraint is

$$F_{max}^2 - (U_1^2 + U_2^2) = U_3^2 \geq 0, \tag{4.29}$$

The second inequality contains maximum rudder angle which is

$$\begin{aligned} \tan^2 \alpha_{max} \geq \tan^2 \alpha = \frac{U_2^2}{U_1^2}, \\ U_1^2 \tan^2 \alpha_{max} - U_2^2 = U_4^2 \geq 0. \end{aligned} \tag{4.30}$$

where U_3 and U_4 are dummy inputs.

Converting the inequalities into equalities, the constraint equations are given by

$$\begin{aligned} C_1 &= U_1^2 + U_2^2 + U_3^2 - F_{max}^2 = 0, \\ C_2 &= U_2^2 + U_4^2 - U_1^2 \tan^2 \alpha_{max} = 0. \end{aligned} \tag{4.31}$$

The control variable vector $\mathbf{z}$ is given by (Figure 2.2)

$$\mathbf{z} = \begin{bmatrix} x_p \\ y_p \end{bmatrix} = \begin{bmatrix} q_1 \\ q_2 \end{bmatrix} \tag{4.32}$$

In discredized model Eq. (4.9) the matrix $\mathbf{f}_k$ is given by

$$\mathbf{f}_k = \mathbf{q}_k + \mathbf{f}(\mathbf{q}_k, \mathbf{U}_k)\Delta t, \tag{4.33}$$

Substituting Eq. (2.26) in $\mathbf{f}(\mathbf{q}_k, \mathbf{U}_k)$

$$\mathbf{f}_k = \mathbf{q}_k + \begin{bmatrix} q_{4_k} \sin q_{3_k} + (q_{5_k} + q_{6_k} d) \cos q_{3_k} \\ q_{4_k} \cos q_{3_k} - (q_{5_k} + q_{6_k} d) \sin q_{3_k} \\ q_{6_k} \\ \frac{1}{m_{11}}(U_{1_k} - d_{11} q_{4_k} + m_{22} q_{5_k} q_{6_k}) \\ \frac{1}{m_{22}}(U_{2_k} - m_{11} q_{4_k} q_{6_k} - d_{22} q_{5_k}) \\ \frac{1}{I_{33}}(m_{11} - m_{22} - L u_{2_k}) q_{4_k} q_{5_k} - d_{66} q_{6_k} \end{bmatrix} \Delta t. \tag{4.34}$$

All the equations are formulation in NMPC derived formulae, for the respective surface vessel. Partial derivatives of these equations are calculated below as

The deriving partial derivatives corresponding to the Eq.s (4.11), (4.12)

$$\frac{\partial \phi}{\partial \mathbf{z}_N^T} = (\mathbf{z}_N - \mathbf{z}_N^d)^T \mathbf{Z}_0, \tag{4.35}$$

$$\frac{\partial L_k}{\partial \mathbf{z}_k^T} = (\mathbf{z}_k - \mathbf{z}_k^d)^T \mathbf{Z}, \tag{4.36}$$

From Eq. (4.32)

$$\frac{\partial \mathbf{z}_k}{\partial \mathbf{q}_k^T} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}^T, \tag{4.37}$$

$$\frac{\partial \phi}{\partial \mathbf{q}_N^T} = (\frac{\partial \phi}{\partial \mathbf{z}_N^T})\,(\frac{\partial \mathbf{z}_N}{\partial \mathbf{q}_N^T}), \tag{4.38}$$

$$\frac{\partial L_k}{\partial \mathbf{q}_k^T} = (\frac{\partial L_k}{\partial \mathbf{z}_k^T})\,(\frac{\partial \mathbf{z}_k}{\partial \mathbf{q}_k^T}). \tag{4.39}$$

The partial derivatives corresponding to these Eq.s (4.34) are

$$\frac{\partial \mathbf{f}_k}{\partial \mathbf{q}_k^T} = \mathbf{I} + \begin{bmatrix} 0 & 0 & A & \mathrm{s}q_{3_k} & \mathrm{c}q_{3_k} & dcq_{3_k} \\ 0 & 0 & B & \mathrm{c}q_{3_k} & -\mathrm{s}q_{3_k} & -dsq_{3_k} \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & \frac{-d_{11}}{m_{11}} & \frac{m_{22}}{m_{11}} q_{6_k} & \frac{m_{22}}{m_{11}} q_{5_k} \\ 0 & 0 & 0 & \frac{-m_{11}}{m_{22}} q_{6_k} & \frac{-d_{22}}{m_{22}} & \frac{-m_{11}}{m_{22}} q_{4_k} \\ 0 & 0 & 0 & \frac{(m_{11}-m_{22})}{I_{33}} q_{5_k} & \frac{(m_{11}-m_{22})}{I_{33}} q_{6_k} & \frac{-d_{66}}{I_{33}} \end{bmatrix} \Delta t, \tag{4.40}$$

where s and c represent sin and cos, respectively, and

$$A = q_{4_k} \mathrm{c} q_{3_k} - (q_{5_k} + d q_{6_k}) \mathrm{s} q_{3_k},$$

$$B = -q_{4_k} \mathrm{s} q_{3_k} - (q_{5_k} + d q_{6_k}) \mathrm{c} q_{3_k}.$$

$$\frac{\partial \mathbf{C}_k}{\partial \mathbf{q}_k^T} = \mathbf{0}. \tag{4.41}$$

The partial derivatives for calculating costates in Eqs. (4.22) and (4.23) are calculated.

The partial derivatives in Eqs. (4.24) and (4.25) are calculated below

$$\frac{\partial L_k}{\partial \mathbf{U}_k^T} = (\mathbf{U}_k - \mathbf{U}_k^d)\mathbf{R} - \mathbf{S}. \tag{4.42}$$

$$\frac{\partial \mathbf{f}_k}{\partial \mathbf{U}_k^T} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ \frac{1}{m_{11}} & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{m_{22}} & 0 & 0 & 0 \\ 0 & -\frac{L}{I_{33}} & 0 & 0 & 0 \end{bmatrix}, \tag{4.43}$$

The partial derivatives corresponding to constraint Eq.s (4.31) are

$$\frac{\partial \mathbf{C}_k}{\partial \mathbf{U}_k^T} = \begin{bmatrix} 2U_1 & 2U_2 & 2U_3 & 0 \\ -2U_1 \tan^2 \alpha_{max} & 2U_2 & 0 & 2U_4 \end{bmatrix}, \tag{4.44}$$

$$\frac{\partial \mathbf{C}_k}{\partial \boldsymbol{\mu}_k^T} = \mathbf{0}, \tag{4.45}$$

These equations are substituted in $6N$ nonlinear equations (4.27) which solves for $4N$ U's for N time steps.

CHAPTER 5

SIMULATION

The model parameters values from the identification tests (Table 3.1) are substituted in Eqs. (2.26) for designing the controller. The length of the distance from CG of the boat to the propellers location on the boat $L = 0.21$. The distance between the CG and CP is assumed to be 1.7 m (Figure 2.2).

The desired trajectory used for tracking is given by

$$x_p^d = Vt, \tag{5.1}$$

$$y_p^d = A(1 - \cos\frac{Vt}{4}). \tag{5.2}$$

Respective velocity components $\dot{x}_p^d$ and $\dot{y}_p^d$ are

$$\dot{x}_p^d = V, \tag{5.3}$$

$$\dot{y}_p^d = \frac{AV}{4}\sin\frac{Vt}{4}. \tag{5.4}$$

where V is the velocity of the boat along the x direction and the velocity of boat along y direction is a sine wave with an amplitude of A. Here the velocity of the boat is taken as $V = 0.9$ m/s and amplitude $A = 1.25$ respectively.

The maximum propeller speed ($n_{\max}$) and rudder angle ($\alpha_{\max}$) attainable by boat are 140 rps and $\pm 30°$. The length of time step is taken to be $\Delta t = 0.25$ sec. The number of time steps of calculation of future control inputs (prediction origin) is $N = 5$. Only the first step of control inputs are applied which is control origin and the calculations are done at each time step.

The equations in the NMPC controller chapter are formulated in MATLAB. The $fsolve$ tool in MATLAB is used to solve system equations ((4.27)) at any control time. These equations are solved simultaneously to find control inputs U_1, U_2 and Lagrange multipliers μ_1, μ_2 for N time steps. The control inputs are substituted in Eqs. (2.26) and intergraded to calculate state space vector which contains position of CP and velocities of CG.

5.1 Non Saturation of Inputs

For testing the controller at non saturation of the inputs, initial position and velocity of the boat are taken same as the desired values. The control gains $\mathbf{Z}_N$, $\mathbf{Z}$, $\mathbf{R}$ and $\mathbf{S}$ are tuned manually using trial and error are given by

$$\mathbf{Z_N} = \begin{bmatrix} 5 \times 10^8 & 0 \\ 0 & 5 \times 10^8 \end{bmatrix}, \quad \mathbf{Z} = \begin{bmatrix} 5 \times 10^8 & 0 \\ 0 & 5 \times 10^8 \end{bmatrix}, \tag{5.5}$$

$$\mathbf{R} = \begin{bmatrix} 5 \times 10^2 & 0 & 0 & 0 \\ 0 & 5 \times 10^2 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad \mathbf{S} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 2 \times 10^4 & 0 \\ 0 & 0 & 0 & 2 \times 10^4 \end{bmatrix}. \tag{5.6}$$

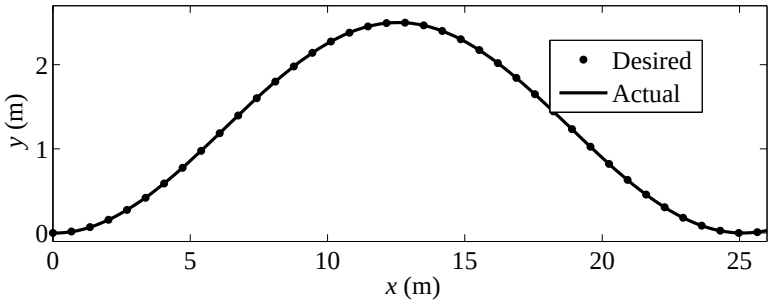


Figure 5.1: Desired and actual path of the control point of the boat for non saturation case

Figure 5.1 shows the path of the control point of the boat. As initial conditions of the boat are same as desired, the perfect tracking of the trajectory is possible right from the start of the test. Figure 5.2 shows the position components of the control point of the boat. The actual x and y coordinates of the boat takes the same shape of the desired position curves (Eqs. (5.1) and (5.2)). The orientation of the boat is not controlled but because of use of control point, it is controlled indirectly. The slope of the tangent to y component curve is zero at the starting point and increases to positive value and decreases back to zero at the top of the curve. It again increases to negative value and decreases to zero. This is exactly reflected in the orientation plot of the boat.

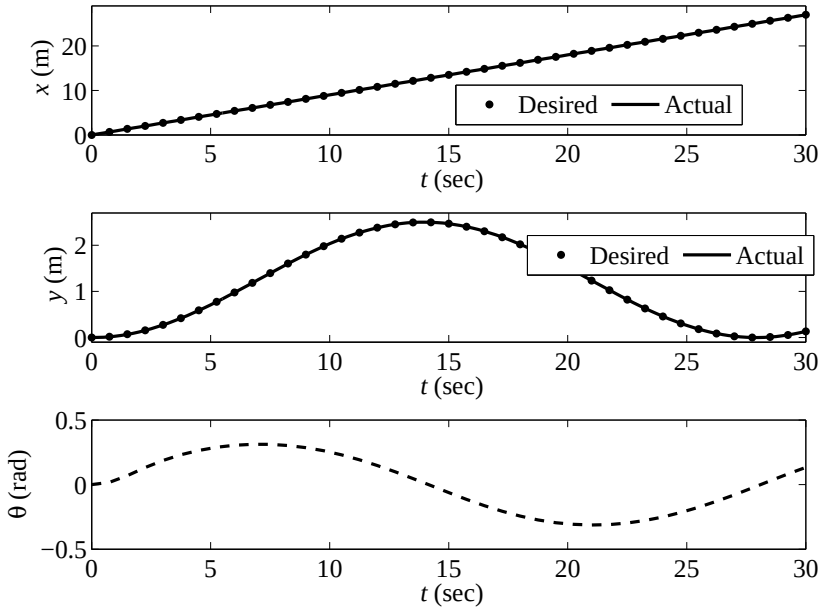


Figure 5.2: The position components of control point and orientation of the boat for non saturation case

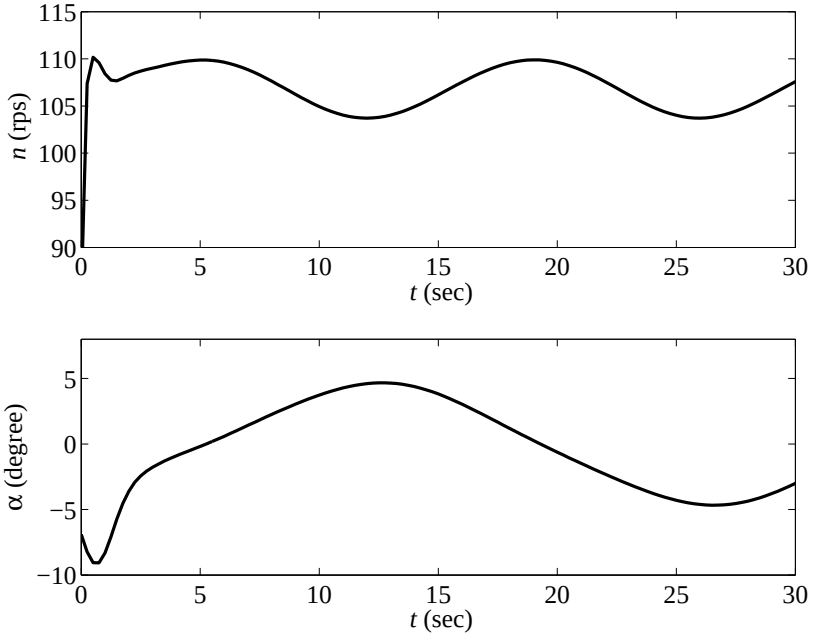


Figure 5.3: The propeller speed and rudder angle of the boat for non saturation case

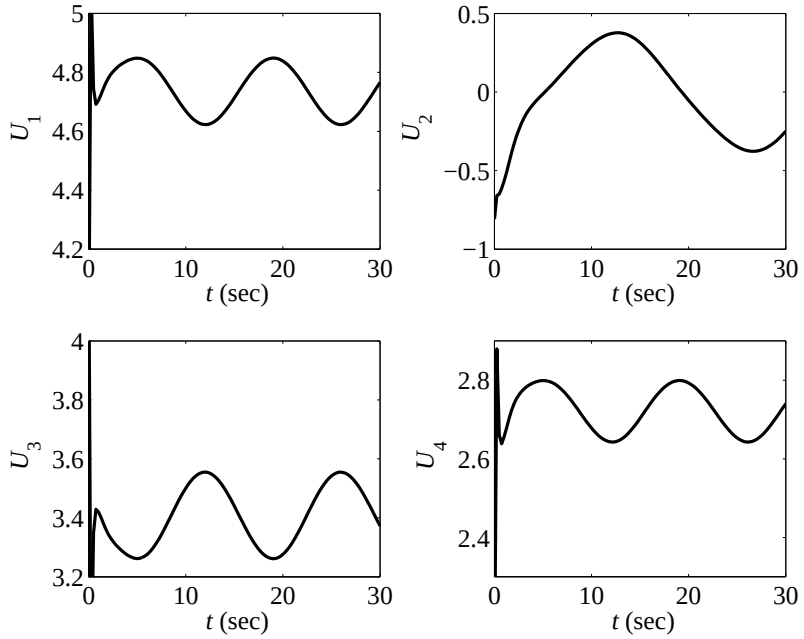


Figure 5.4: The control inputs and dummy inputs for the boat for non saturation case

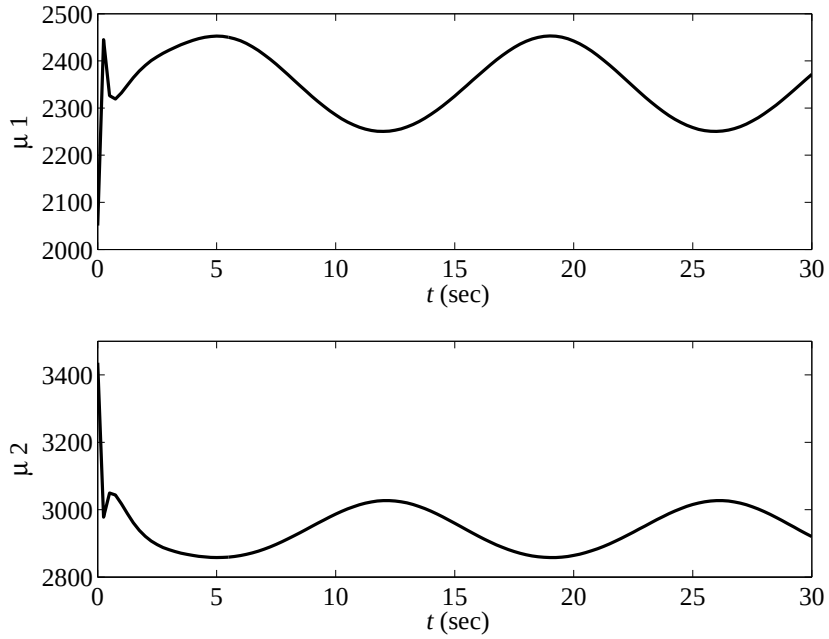


Figure 5.5: The Lagrange multipliers for non saturation case

Figure 5.3 shows the propeller speed and rudder angle of the boat and both the values are within the maximum limit. Figure 5.4 shows the control inputs and dummy inputs applied to the boat. The control input u_1 and u_2 takes the shape of n and α curves which are physical control inputs of the boat. The values of control inputs are very less because initial conditions same as desired and no extra input needed to cover the offset. The dummy inputs u_3 and u_4 represent the offset of control inputs from maximum inputs. The values of u_3 and u_4 are high and non zero which shows well within the maximum limits. Figure 5.5 shows the Lagrange multipliers, which are for penalizing the constraint equations and these values are also low because of non saturation of inputs.

5.2 Saturation of Inputs

For testing the controller at saturation of the inputs, initial conditions of the boat are taken not same as the desired values. As the initial conditions are not same as desired and far away, the inputs of the boat saturate to the maximum values and the performance of the NMPC is tested. These are the usual initial conditions in testing boat experimentally. The control gains $\mathbf{Z}_N$, $\mathbf{Z}$, $\mathbf{R}$ and $\mathbf{S}$ in Eqs. (6.13) and (5.6) are used.

Figure 5.6 and Figure 5.7 shows the boat is at (0,-1) initially and initial orientation of the boat is -15°. Figure 5.6 and Figure 5.7 shows the convergence of the path of the control point of the boat to the desired path after 13 sec and it is perfect tracking from then. The velocity of control point of the boat is calculated using u, v

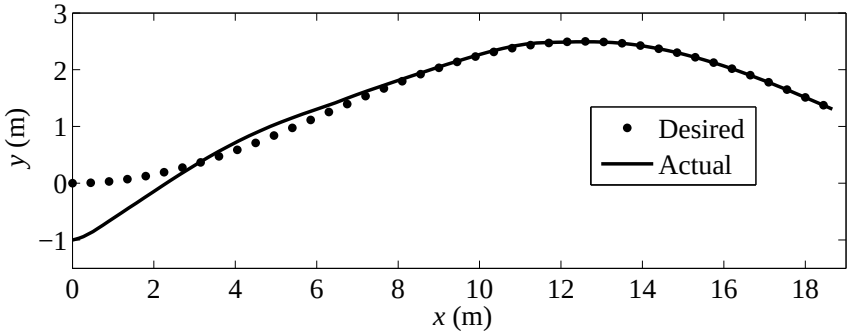


Figure 5.6: Desired and actual path of the control point of the boat for saturation case

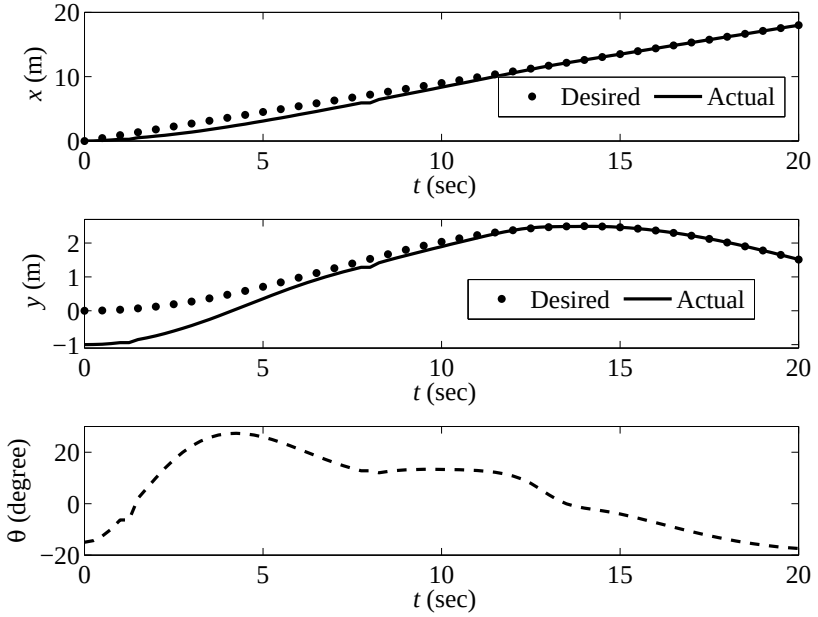


Figure 5.7: The position components of control point and orientation of the boat for saturation case

and r values is given by

$$u_p = u, \quad v_p = v + rd, \tag{5.7}$$

$$V = \sqrt{{u_p}^2 + {v_p}^2}. \tag{5.8}$$

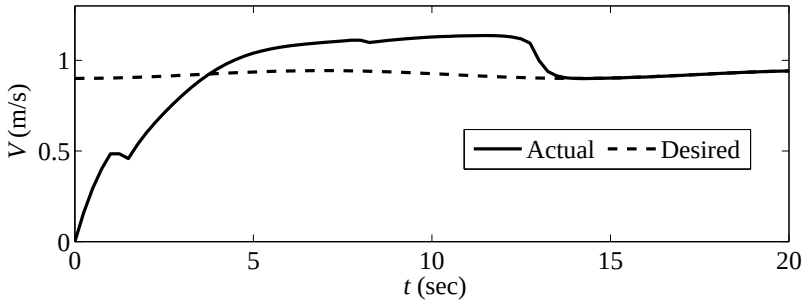


Figure 5.8: Desired and actual velocity of the boat for saturation case

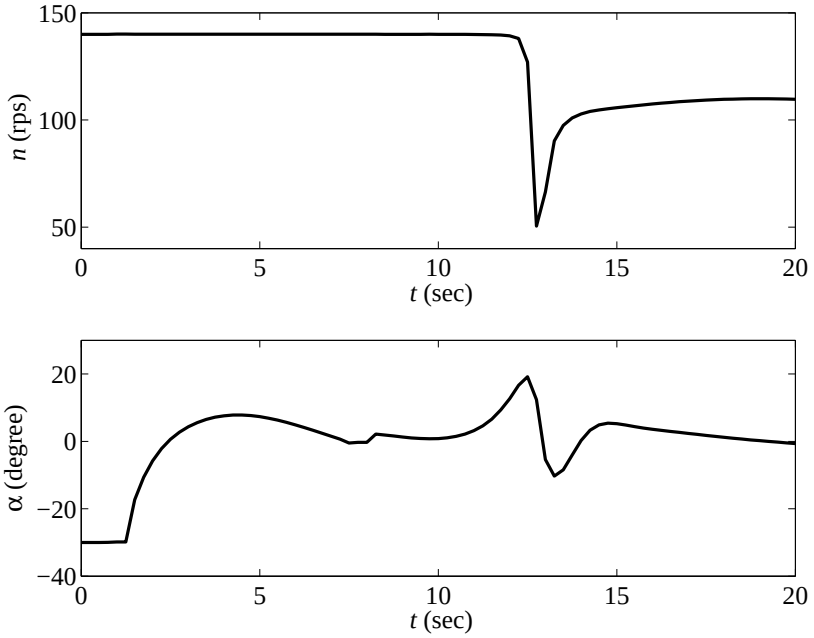


Figure 5.9: The propeller speed and rudder angle of the boat for saturation case

where u_p and v_p are longitudinal and lateral velocities of control point of the boat. Figure 5.8 shows the initial velocity of the boat is zero and it rises and overshoots the desired value $V = 0.9$ m/s to catch up the desired path till 13 sec and reaches to the constant desired velocity.

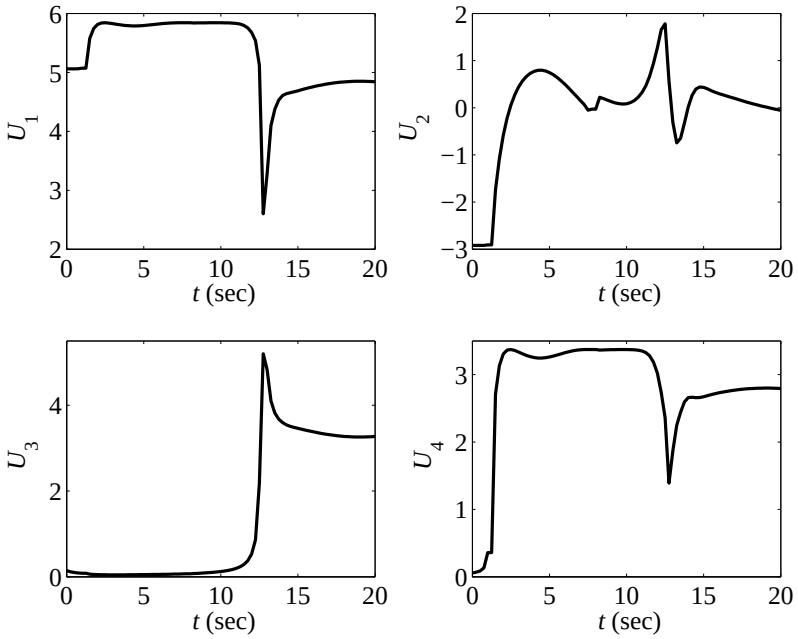


Figure 5.10: The control inputs and dummy inputs for the boat for saturation case

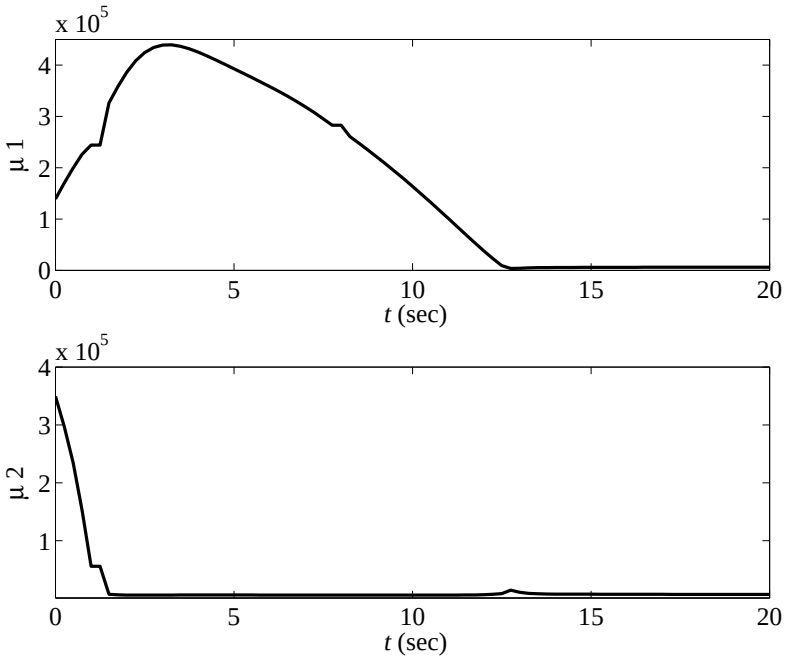


Figure 5.11: The Lagrange multipliers for saturation case

Figure 5.9 shows the propeller speed reaches the maximum value and stays constant till it converges to the desired path. The small spike is due to sudden change in propeller speed to normal desired speed. As the initial orientation of the boat is -15°, the rudder angle of boat also saturates for small time until its course is parallel to the desired. Figure 5.10 shows the control inputs are high for all the time of saturation of inputs. The dummy inputs stays zeros i.e offset between maximum and actual inputs is zero for all the saturation time and reaches a constant value after 13 sec. Figure 5.11 shows the Lagrange multiplier values are very high to penalize the constraint equations and therefore constraining inputs not going beyond the maximum limits. After 13 sec these values are very less showing the no need for penalizing inputs.

The performance of the controller through these simulations seems to be very impressive, so the controller can be tested experimentally on the real robotic boat.

CHAPTER 6

EXPERIMENTATION

For the real time application of NMPC, the time for solving the $6N$ nonlinear equations (4.27) at each time step must be small and the accuracy of solution is crucial. The solver used for simulations does not have time limits. In real time applications due to the disturbances the convergence of error in minimum iterations over the time is not possible if the solution method is not accurate. The calculation time also has to be constant inspite of any extra variation of solution situation over the whole control time. Hence the solution method must be selected such that it has small and constant calculation time and is accurate. One of such solution methods is Generalized Minimum RESidual method (GMRES) along with continuation method.

6.1 The Continuation Method

The system of $6N$ nonlinear equations (4.27) are converted into differential equations whose equilibrium point constitutes the solution $\mathbf{u}$ for these nonlinear equations.

$$\dot{\mathbf{F}}(\mathbf{u}(t), \mathbf{q}(t), t) = \mathbf{A}_s \mathbf{F}(\mathbf{u}(t), \mathbf{q}(t), t), \tag{6.1}$$

where $\mathbf{A}_s$ is a negative definite matrix and initial solution $\mathbf{u}_0$ of $\mathbf{F}(\mathbf{u}_0, \mathbf{q}(0), 0) = \mathbf{0}$ makes a stable system.

$\dot{\mathbf{F}}$ is calculated

$$\dot{\mathbf{F}} = \mathbf{F}_u \dot{\mathbf{u}} + \mathbf{F}_q \dot{\mathbf{q}} + \mathbf{F}_t, \tag{6.2}$$

Substituting in Eq. (6.1) and solving for $\dot{\mathbf{u}}$

$$\dot{\mathbf{u}} = \mathbf{F}_u^{-1}(\mathbf{A}_s \mathbf{F} - \mathbf{F}_q \dot{\mathbf{q}} - \mathbf{F}_t). \tag{6.3}$$

where

$$\mathbf{F}_u = \frac{\partial \mathbf{F}}{\partial \mathbf{u}^T}, \quad \mathbf{F}_q = \frac{\partial \mathbf{F}}{\partial \mathbf{q}^T}, \quad \mathbf{F}_u = \frac{\partial \mathbf{F}}{\partial t}. \tag{6.4}$$

Note that the closed form of the function $\mathbf{F}$ is not available. So, the above partial derivatives can not be calculated. Therefore $\dot{\mathbf{u}}$ can not be determined by using Eq. (6.3). To address that problem the GMRES method is used to solve for $\dot{\mathbf{u}}$ iteratively and is integrated in time to solve for control inputs and Lagrange multipliers. The approach is presented in the next section.

6.2 The Generalized Minimum Residual Method

In mathematics, the GMRES is an iterative method for the numerical solution of system of equations. This method obtains an approximate solution from the Krylov subspace with minimal residual. The Arnoldi iteration is used to find this vector. This

method is used to solve for $\dot{\mathbf{u}}$ at each time step such that the residual r is minimized.

$$r = ||\mathbf{F}_u\dot{\mathbf{u}} - (\mathbf{A}_s\mathbf{F} - \mathbf{F}_q\dot{\mathbf{q}} - \mathbf{F}_t)||. \tag{6.5}$$

GMRES converges to the exact solution minimizes the residual norm after $6N$ iterations which is equal to the number of equations but even after less iterations relative to $6N$, the solution is very close to the exact solution. For practical applications, a fixed less number of iterations can be specified which not only reduces the calculation time but also guarantees a constant calculation time thus maintaining the calculation burden constant all the time for the feasibility of real time controllers. In some practical applications, even the maximum number of iterations to arrive at the exact solution is practically feasible.

The rate of control inputs and Lagrange multipliers $\dot{\mathbf{u}}$ at each time step are integrated to solve for corresponding control step heights and Lagrange multipliers. GMRES needs an initial guess of $\mathbf{u_{0g}}$ and $\dot{\mathbf{u}}_{\mathbf{0g}}$ for the first iteration of calculation at the first time step. For initial guess $\mathbf{u}$ at other time steps, the solution from the previous control steps is used. The initial guess $\dot{\mathbf{u}}$ is always zero. The control inputs at each time step are calculated as

$$\mathbf{u}_0 = \mathbf{u}_{0g} + \dot{\mathbf{u}}_0 \Delta t, \tag{6.6}$$

$$\mathbf{u}_k = \mathbf{u}_{k-1} + \dot{\mathbf{u}}_k \Delta t, \quad k = 1, ...N-1 \tag{6.7}$$

$$\boldsymbol{\mu}_0 = \boldsymbol{\mu}_{0g} + \dot{\boldsymbol{\mu}}_0 \Delta t, \tag{6.8}$$

$$\boldsymbol{\mu}_k = \boldsymbol{\mu}_{k-1} + \dot{\boldsymbol{\mu}}_k \Delta t, \quad k = 1, ...N-1. \tag{6.9}$$

$$\tag{6.10}$$

For calculating analytically $\mathbf{F}_u$ $(6N \times 6N)$ and $\mathbf{F}_q$ $(6N \times 5N)$ in Eq. (6.3) to solve for $\dot{\mathbf{u}}$ is very difficult. Even the approximation to these matrices is very cumbersome to compute due to its large size, hence an approximation of these with the product of the $\mathbf{F}_u \dot{\mathbf{u}}$ and $\mathbf{F}_q \dot{\mathbf{q}} + \mathbf{F}_t$ is simple to find using the finite difference method.

$$\mathbf{F}_u \dot{\mathbf{u}} \approx \frac{\mathbf{F}(\mathbf{u} + \dot{\mathbf{u}}\Delta t, \mathbf{q}, t) - \mathbf{F}(\mathbf{u}, \mathbf{q}, t)}{\Delta t}, \tag{6.11}$$

$$\mathbf{F}_q \dot{\mathbf{q}} + \mathbf{F}_t \approx \frac{\mathbf{F}(\mathbf{u}, \mathbf{q} + \dot{\mathbf{q}}\Delta t, t + \Delta t) - \mathbf{F}(\mathbf{u}, \mathbf{q}, t)}{\Delta t}. \tag{6.12}$$

With these approximations and using GMRES method, a solver is developed which solves for $\dot{\mathbf{u}}$ based on specified residual norm r and number of iterations.

6.3 Experimental Results

The closed loop NMPC controller programs are tested in an outdoor pond. The controller is tested for tracking straight path and a sine path with different gains. A constant longitudinal velocity of 0.8 m/s is specified right from the starting point. Four sets of tests are performed and for each test the boat is driven manual to the center of the pond and controller is triggered when the boat is almost at rest. The control gains used for each test are given in Table 6.1. The outdoor pond has a fountain at the other end of the pond which generates disturbance waves which causes a lateral disturbances of 0.2 m/s velocity approximately. Based on the maximum propeller speed, the maximum attainable velocity of the boat is 1 m/s. This causes a 20% disturbances on the boat. While tracking the longitudinal velocity of 0.8 m/s, the boat has to compensate for these lateral disturbances. In vicinity of disturbances, the boat is pushed to its limits and this causes a slow convergence of boat on the desired path. Two straight line and two sine path tests are done. For all tests, a tolerance of convergence for $\dot{\mathbf{u}}$ i.e $|\dot{\mathbf{u}}_{\mathbf{k+1}} - \dot{\mathbf{u}}_{\mathbf{k}}| < 10^{-25}$ and a fixed iterations of 30 are specified. These constraints makes sure that the error in $\dot{\mathbf{u}}$ calculation is minimum and the fixed iteration number is equal to number of unknowns in the equations the GMRES has to solve, the main advantage of GMRES is, at the iteration number of total unknowns in the equations, at this point the solution of $\dot{\mathbf{u}}$ is exact. For forcing the GMRES to reach 30 iterations a very low residual is defined such that every time the solver quits at 30 iterations giving an exact solution.

The control gains used for each test are

$$\mathbf{Z_N} = \quad \mathbf{Z} = \begin{bmatrix} A & 0 \\ 0 & A \end{bmatrix}, \tag{6.13}$$

$$\mathbf{R} = \begin{bmatrix} B & 0 & 0 & 0 \\ 0 & B & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad \mathbf{S} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & C & 0 \\ 0 & 0 & 0 & C \end{bmatrix}. \tag{6.14}$$

Table 6.1: Control gains

Test	A	B	C
1.	1.2×10^7	10^3	10^4
2.	1.2×10^7	10^3	10^4
3.	1.5×10^7	10^3	10^4
4.	1.5×10^7	10^3	10^4

6.3.1 Test 1

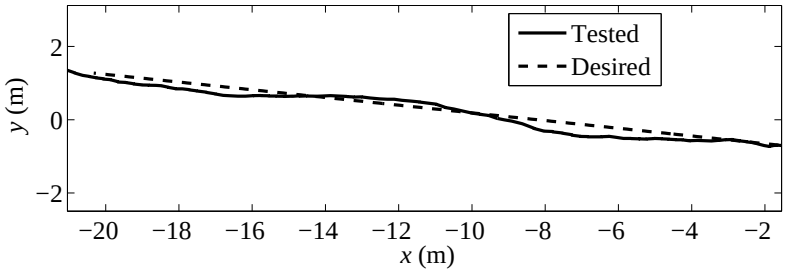


Figure 6.1: The trajectory tracking of the control point of the boat in real time with respect the desired path for the test1

This is a straight line test with constant desired velocity of boat in x direction equal to 0.8 m/s. Figure 6.1 shows the actual path of the boat on the desired path.

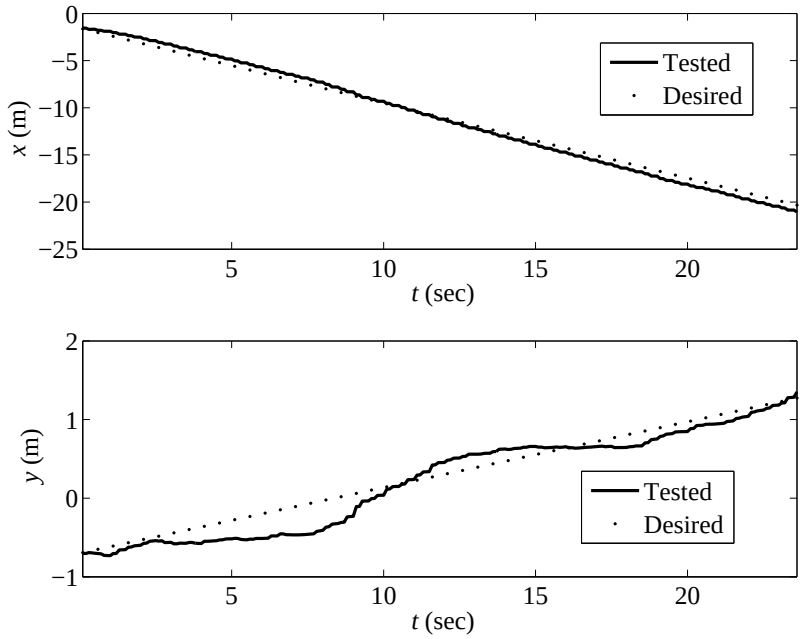


Figure 6.2: The x and y coordinates of the control point of the boat in real time for test 1

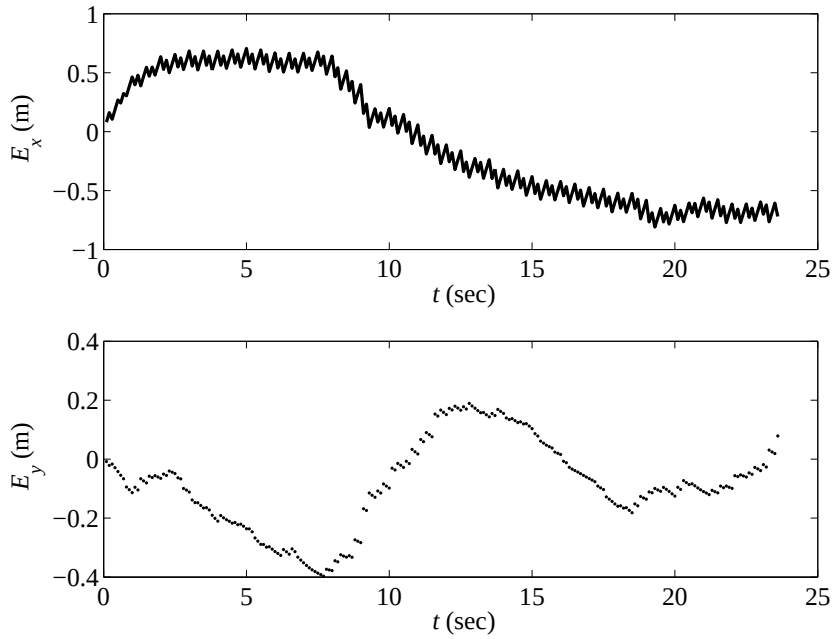


Figure 6.3: The error in actual and desired x and y coordinates for test 1

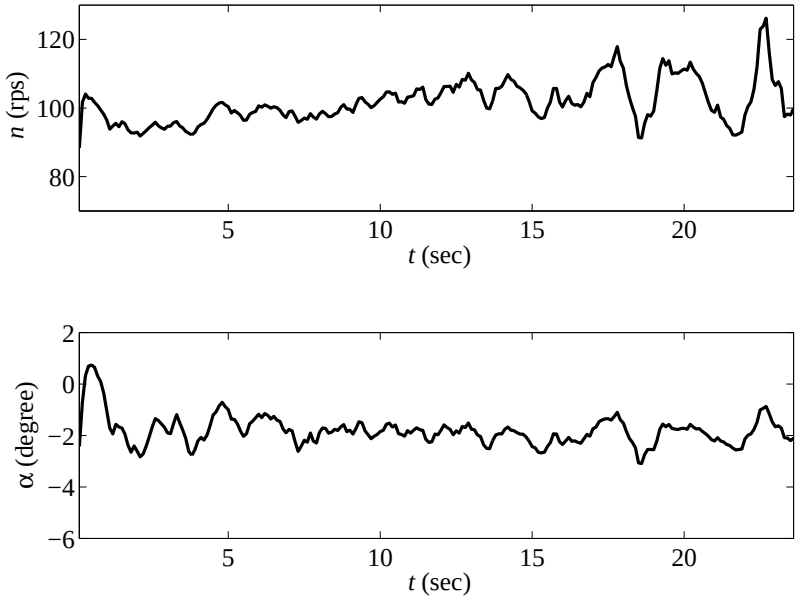


Figure 6.4: The propeller speed and rudder angle of the boat in real time for test 1

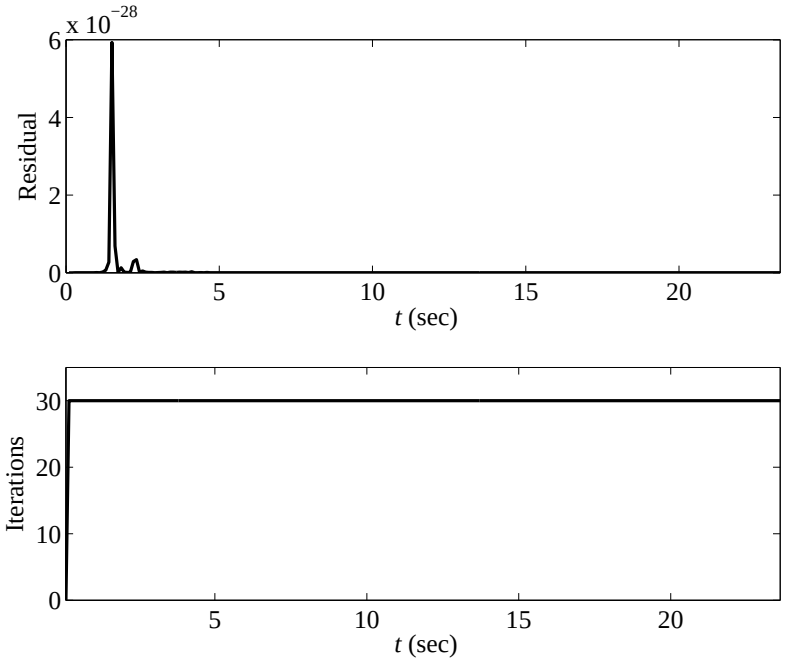


Figure 6.5: The residual in $\dot{\mathbf{u}}$ calculation and number of iterations taken for test1

Figure 6.2 shows boat starts at (-1,-1) which is same as the desired. From these two figures the boat is following the desired path accurately and in case of any lateral disturbance pushing it off the path, its trying to converge on to the desired path. Figure 6.3 shows the error in x and y coordinates are at most 1 m which is below the position accuracy of the sensor 2.5 m (Figure 8.2), so these errors in x and y coordinates are due to error in sensor readings, this proves the perfect tracking of the desired path. To keep the boat on course the rudder angle α which rather being zero, maintains a constant negative angle of -2° Figure 6.4 to compensate for lateral disturbances. The propeller speed (n) rises initially to catch up the desired velocity and it stays constant, as the boat approaches the fountain the disturbances on the boat increases which causes the fluctuations in n values. Figure 6.5 shows the residual r is very low and GMRES takes fixed number of iterations $6N$ (30) for solving, thus guaranteeing the exact solution.

6.3.2 Test 2

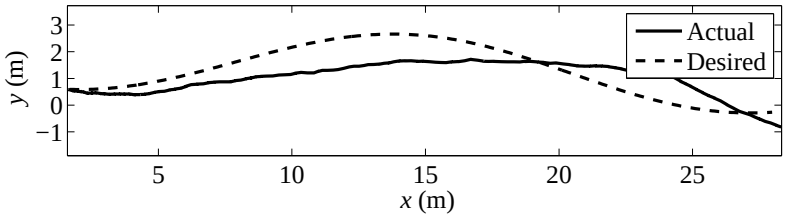


Figure 6.6: The trajectory tracking of the boat in real time with respect the desired path for test 2

This test is sine path tracking with constant desired velocity of boat in x direction of 0.8 m/s. The control gains are same as test 1. Figure 6.6 shows the

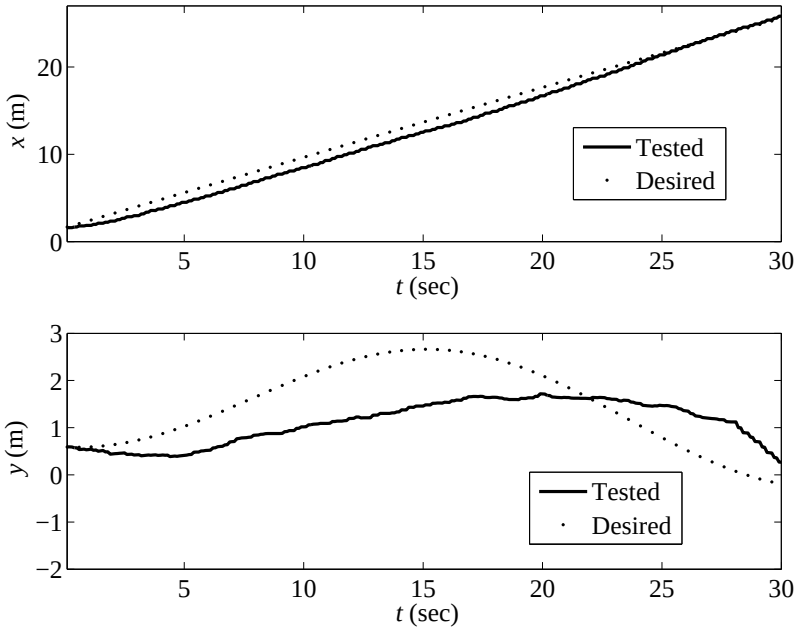


Figure 6.7: The x and y coordinates of the control point of the boat in real time for test 2

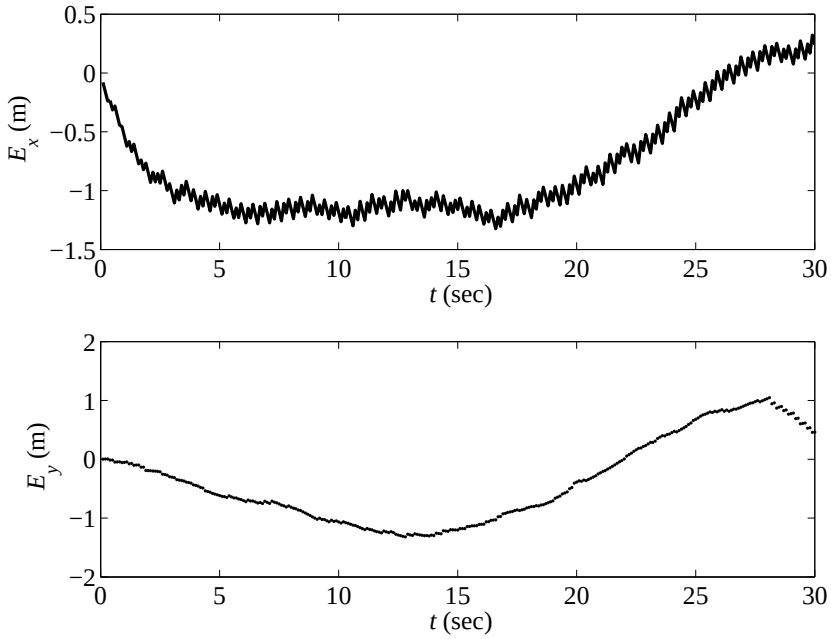


Figure 6.8: The error in actual and desired x and y coordinates for test 2

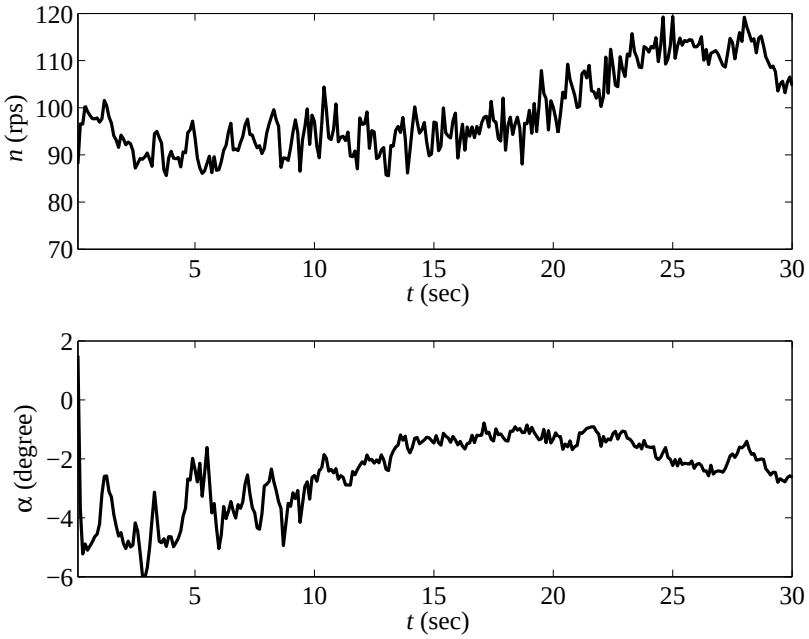


Figure 6.9: The propeller speed and rudder angle of the boat in real time for test 2

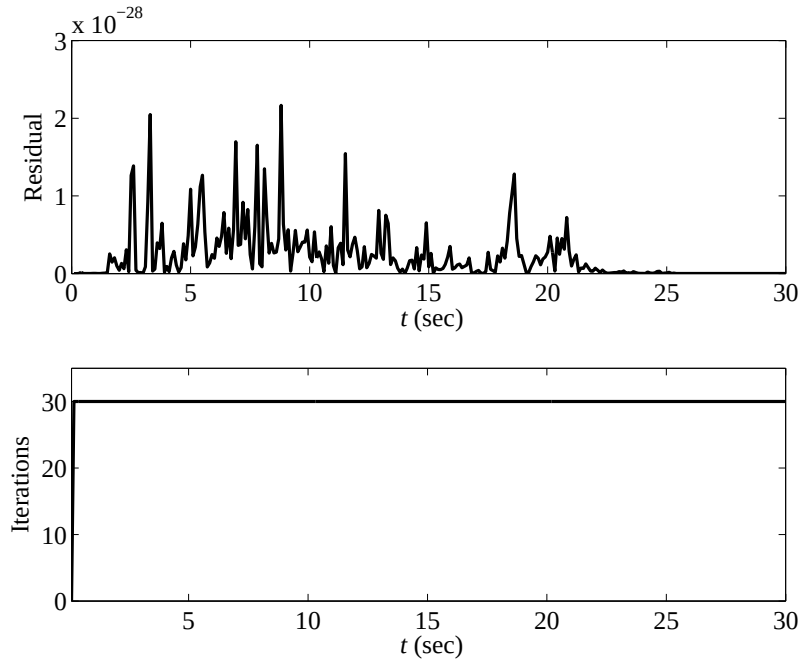


Figure 6.10: The residual in $\dot{\mathbf{u}}$ calculation and number of iterations taken for test 2

actual path of the boat on the desired path which is tracking the path and slowly converging on the desired. Figure 6.7 boat starts at (1,0.5) which is same as the desired. Figure 6.8 shows the error in x and y coordinates are at most 1.5 m which is below the position accuracy of the sensor 2.5 m (Figure 8.2), so these errors in x and y coordinates are due to error in sensor readings and also due to the lateral disturbance which making the convergence slow. Figure 6.9 shows n value rises slowly and the α adjusting converge on to the path. Figure 6.10 shows the residual r is very low and GMRES takes fixed number of iterations $6N$ (30) for solving, thus guaranteeing the exact solution.

6.3.3 Test 3

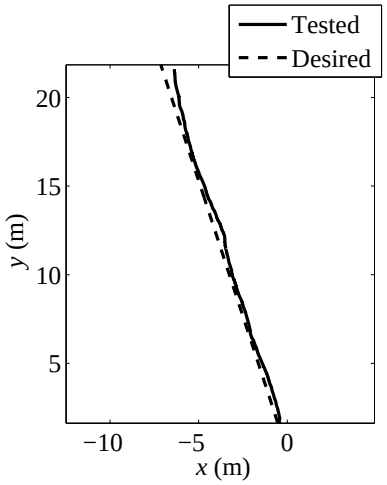


Figure 6.11: The trajectory tracking of the boat in real time with respect the desired path for test 3

This is a straight line test with constant desired velocity of boat in x direction equal to 0.8 m/s. Figure 6.11 shows the actual path of the boat on the desired path.

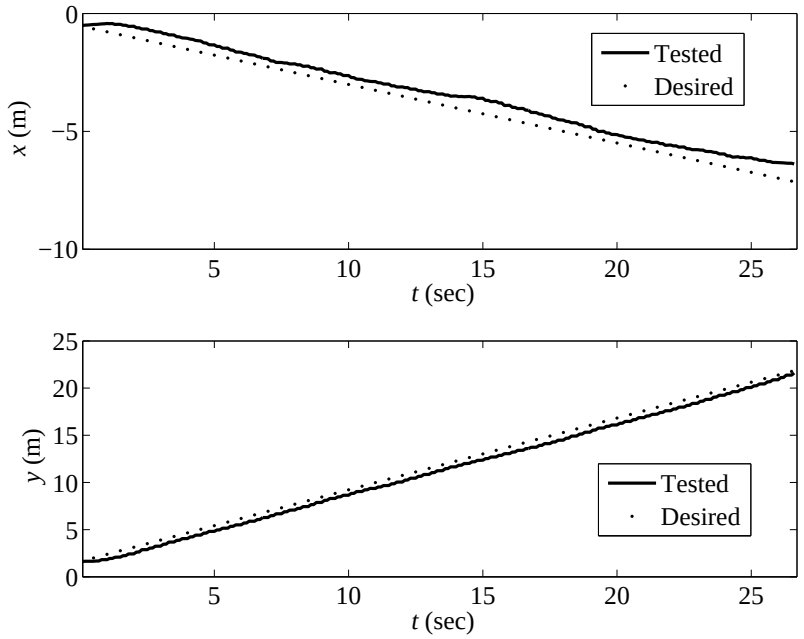


Figure 6.12: The x and y coordinates of the control point of the boat in real time for test 3

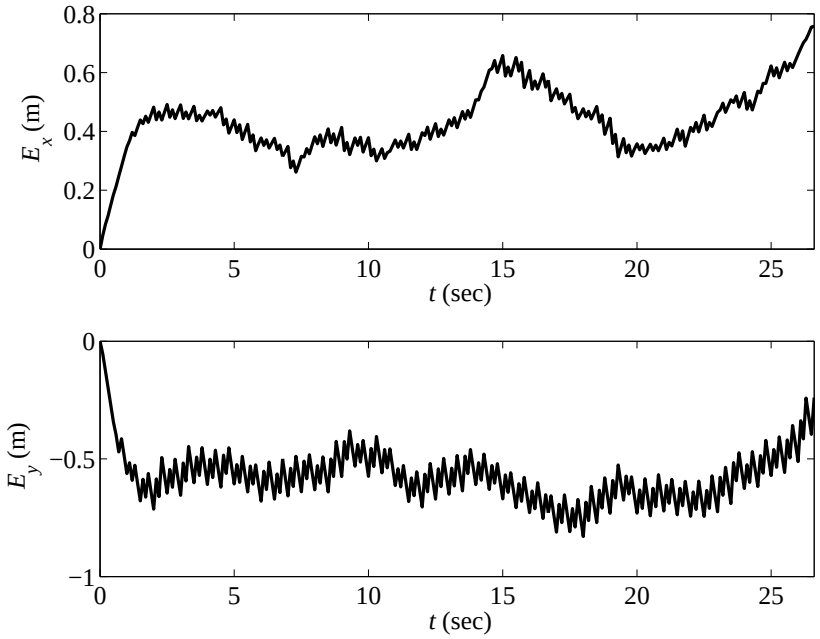


Figure 6.13: The error in actual and desired x and y coordinates for test 3

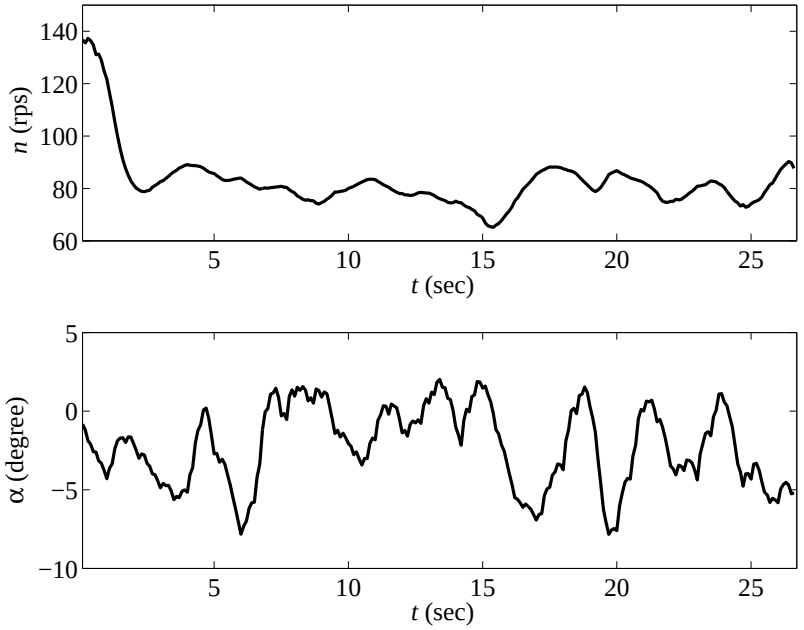


Figure 6.14: The propeller speed and rudder angle of the boat in real time for test 3

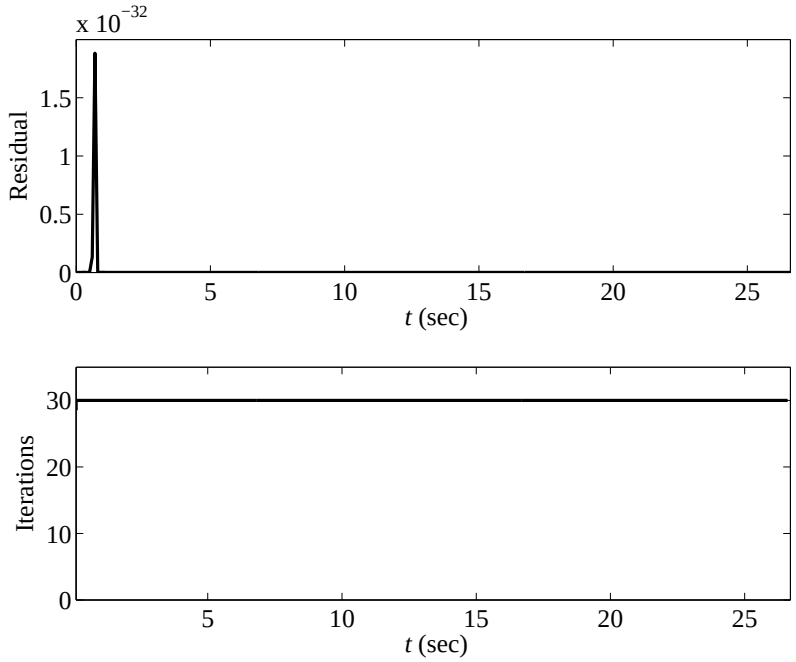


Figure 6.15: The residual in $\dot{\mathbf{u}}$ calculation and number of iterations taken for test 3

Figure 6.12 shows boat starts at (-0.5,1) which is same the desired. From these two figures the boat is following the desired path accurately with some small constant error between the desired. Figure 6.13 shows the error in x and y coordinates are at most 0.75 m which is below the position accuracy of the sensor 2.5 m (Figure 8.2), so these errors in x and y coordinates are due to error in sensor readings, this proves the perfect tracking of the desired path. Figure 6.14 shows the initial high propeller speed n to catch the desired path and after drops down and stays constant at 80 rps. The rudder angle α of the boat is close to zero but the small fluctuations are due to the controller correcting the boat to keep on course. Figure 6.15 shows the residual r is very low and GMRES takes fixed number of iterations $6N$ (30) for solving, thus guaranteeing the exact solution.

6.3.4 Test 4

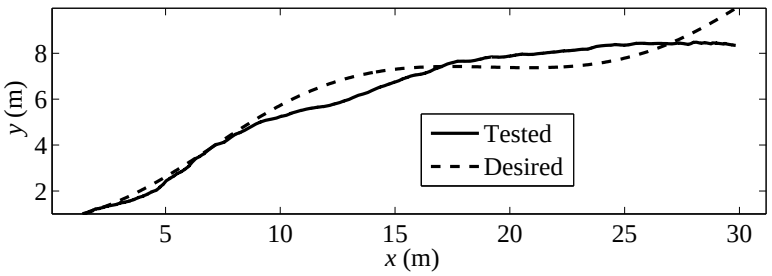


Figure 6.16: The trajectory tracking of the boat in real time with respect the desired path for test 4

This test is sine path tracking with constant desired velocity of boat in x direction of 0.8 m/s. The control gains are same as test 3. Figure 6.16 shows the actual path of the boat on the desired path which is tracking the path and slowly

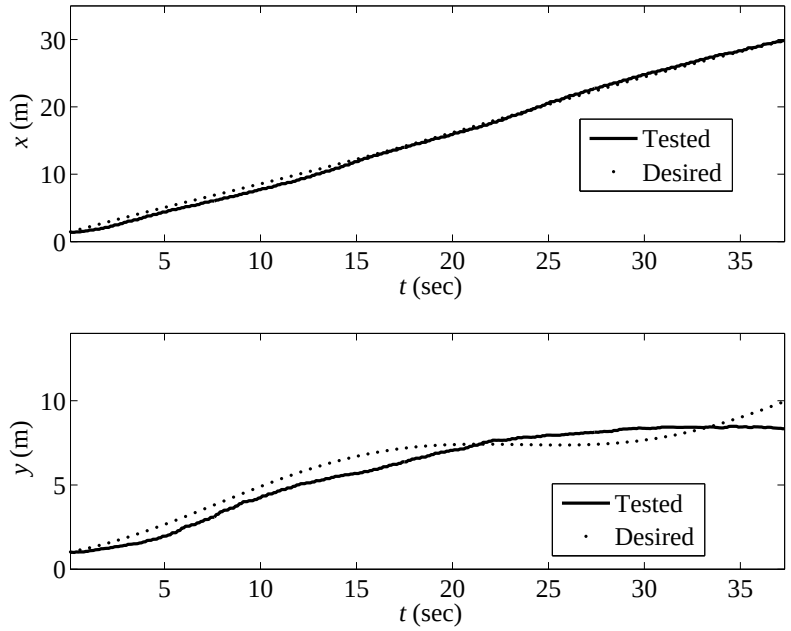


Figure 6.17: The x and y coordinates of the control point of the boat in real time for test 4

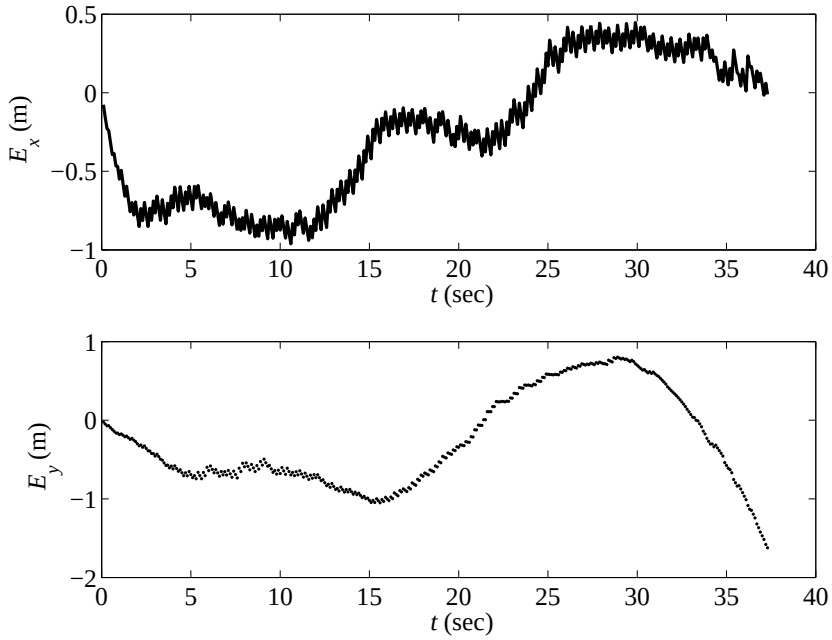


Figure 6.18: The error in actual and desired x and y coordinates for test 4

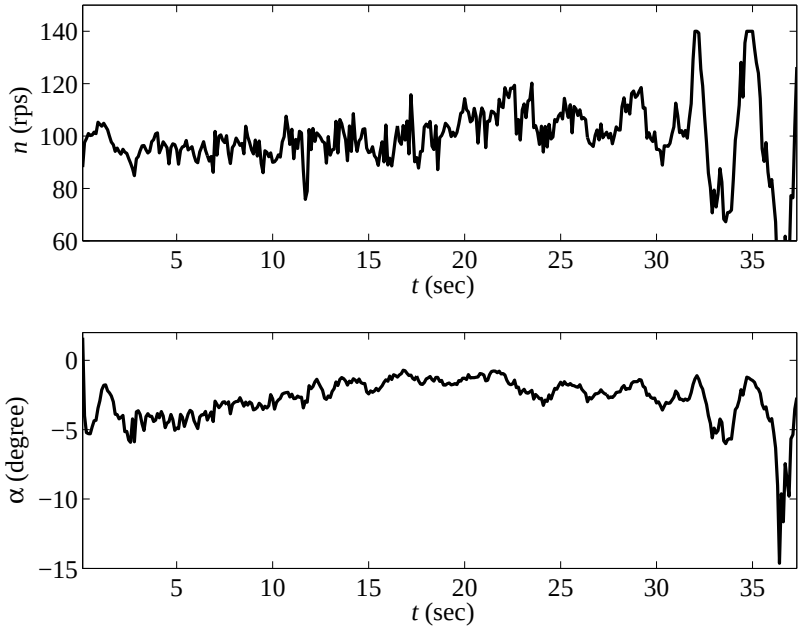


Figure 6.19: The propeller speed and rudder angle of the boat in real time for test 4

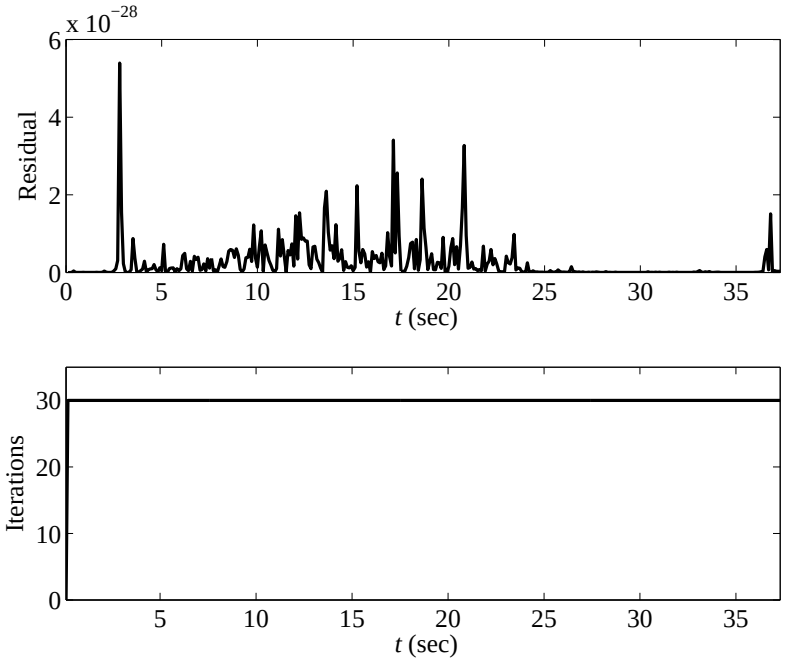


Figure 6.20: The residual in $\dot{\mathbf{u}}$ calculation and number of iterations taken for test4

converging on the desired. Figure 6.17 boat starts at (2,1) which is same as the desired. Figure 6.18 shows the error in x and y coordinates are at most 1 m except at the end which is below the position accuracy of the sensor 2.5 m (Figure 8.2), so these errors in x and y coordinates are due to error in sensor readings and also due to the lateral disturbance which making the convergence slow. Figure 6.19, the rise in n values stays constant to reach the desired longitudinal velocity and the rudder angle α is initially high to drive the boat on to the desired and in both the plots at the end as the boat approaches the fountain the disturbances causes the boat to diverge in y direction (Figure 6.17) which causes high fluctuations in n and α. Figure 6.20 shows the residual r is very low and GMRES takes fixed number of iterations $6N$ (30) for solving, thus guaranteeing the exact solution.

CHAPTER 7

CONCLUSION

A Nonlinear Model Predictive Controller for the trajectory tracking with incorporated input saturation limits is proposed and applied to a 3 DOF underactuated surface vessel. Underactuated surface vessels due to the lack of holonomic constraints and number of control inputs (2) less than the state variables, it is difficult to control the CG of the vessels. Here a better approach of controlling control point (CP) of the boat is presented and results shows that its tracking performance is increased when compared with control of CG. The common practical problem of input saturation in all dynamic systems is considered in the design of the NMPC. For designing a accurate controller, identification tests for the estimation of dynamic model parameters are performed and verified. The performance of the controller is tested through the simulation and experimentation. A desired trajectory of sine path with a constant velocity in x direction is defined. The simulations results for input saturation and a normal case are presented. The initial conditions of the boat which are not same as desired are specified to test the performance of the vessels at the input saturation. The results for the input saturation shows that the control inputs stays within the maximum bounded values until it converges to the desired path. In case of nor-

mal situation, the controller shows the exact tracking right from the start without any saturation. The experimentation results of NMPC in outdoor pond using a small robotic boat are presented. For real time optimization of the NMPC, GMRES method which has constant and less calculation time is presented. Experimental results for a straight path and sine path are presented. These results showed the performance of the controller is good.

CHAPTER 8

APPENDIX

8.1 Hardware Setup

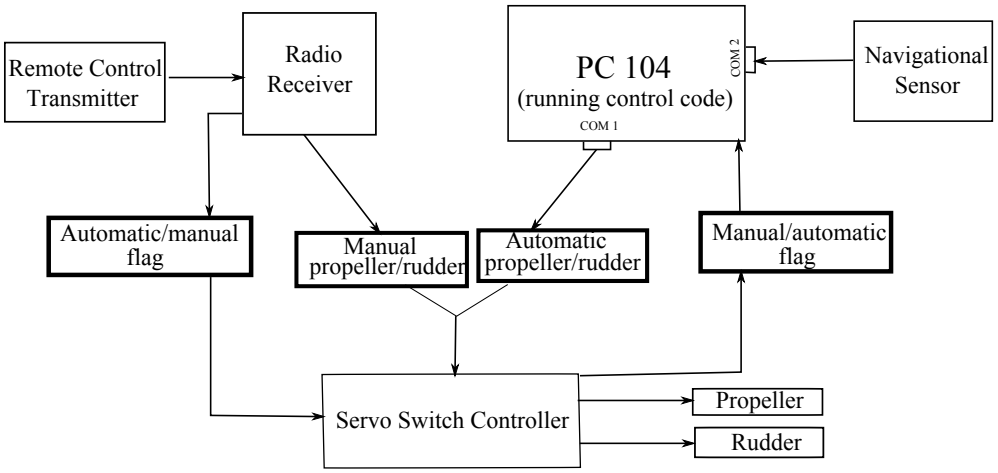


Figure 8.1: The hardware setup inside the control box

Figure 8.1 show the hardware setup inside the control box. It contains a PC 104, navigational sensor, receiver and servo switch card. The PC 104 is a onboard real time computer which runs an embedded program inside. The PC 104 has two serial port COM1 and COM2 which receive and transfer data. The navigational sensor outputs all 6 DOF of the boat and their respective derivatives which are fed to the PC 104 via COM2 port. The GPS values of the sensor are in Geodetic system which

are converted to local coordinated using the appropriate transformations [52]. During the manual mode the servo switch transfers remote control signals to control inputs. In the automatic mode the PC 104 computes the required control inputs based on the sensor readings and send to the servo switch which feeds control inputs to the servo motors on the boat. The power for the servo motors on the boat is supplied through two 7.2 V batteries. The PC 104 and sensor are powered separately with two 11.1 V batteries. The servo switch gets the power from the PC 104.

8.2 Hardware Configuration

Base Remote Control System: Traxxas Villain Ex Deepv RTR Racer 14.4V with Rudder and Thruster Control.

On – Board Control Computer: 1.8Ghz Pentium M PC104 Industrial Computer, 40GB HD, 1GB RAM, 1GB Flash Drive, 2 Ethernet ports, 2 Serial ports, On-board Graphics.

Actuator Controls: Dual-Mode (remote/computer) Controller Board Accepting Servo Command via Serial port or Radio Receiver.

Navigational Sensor: *CrossBow NAV*440 – 6 DOF IMU with 3 - Axis Internal Magnetometer, and Internal WAAS Capable GPS Receiver Position, Dynamic

Performance	NAV440
Position/Velocity	
Position Accuracy[1] (m CEP)	< 2.5
X,Y Velocity Accuracy (m/s rms)	0.4
Z Velocity Accuracy (m/s rms)	0.5
1PPS Accuracy (ns)	± 50
Heading	
Range (°)	± 180
Accuracy[2, 3] (°)	< 1.0
Resolution (°)	< 0.02
Attitude	
Range: Roll, Pitch (°)	± 180, ± 90
Accuracy[2, 3] (°)	< 0.2
Resolution (°)	< 0.02
Angular Rate	
Range: Roll, Pitch, Yaw (°/sec)	± 200 (± 400 option available)
Bias Stability In-Run[4] (°/hr)	< 10
Bias Stability Over Temp[2] (°/sec)	< 0.02
Resolution (°/sec)	< 0.02
Bandwidth (Hz)	25

Figure 8.2: The performance data sheet of NAV440 sensor

Velocity, and Dynamic Roll, Pitching Heading.

Software: MATLAB/SIMULINK/xPC-Target Fast Control Prototyping System

REFERENCES

[1] C. Kitts, P. Mahacek, T. Adamek, and I. Mas. Experiments in the control and application of automated surface vessel fleets. In *OCEANS 2011*, pages 1 –7, sept. 2011.

[2] So-Ryeok Oh and Jing Sun. Path following of underactuated marine surface vessels using line-of-sight based model predictive control. *Ocean Engineering*, 37(2-3):289 – 295, 2010.

[3] Geir Beinset and Jarle Saga Blomhoff. Controller design for an unmanned surface vessel : Design of a heading autopilot and way-point navigation system for an underactuated usv., 2007.

[4] K.Y. Pettersen and E. Lefeber. Way-point tracking control of ships. In *Decision and Control, 2001. Proceedings of the 40th IEEE Conference on*, volume 1, pages 940 –945 vol.1, 2001.

[5] A.J. Sorensen, S.I. Sagatun, and T.I. Fossen. Design of a dynamic positioning system using model-based control. *Control Engineering Practice*, 4(3):359 – 368, 1996.

[6] T. Holzhueter. LQG approach for the high-precision track control of ships. *IEE Proceedings: Control Theory and Applications*, 144(2):121 – 127, 1997.

[7] L. Morawski and J. Pomirski. Ship track-keeping: Experiments with a physical tanker model. *Control Engineering Practice*, 6(6):763 – 769, 1998.

[8] J.-M. Godhavn, T.I. Fossen, and S.P. Berge. Non-linear and adaptive backstepping designs for tracking control of ships. *International Journal of Adaptive Control and Signal Processing*, 12(8):649 – 670, 1998.

[9] K.Y. Pettersen and H. Nijmeijer. Global practical stabilization and tracking for an underactuated ship - a combined averaging and backstepping approach. *Modeling, Identification and Control*, 20(4):189 – 199, 1999.

[10] Kristin Y. Pettersen and H. Nijmeijer. Underactuated ship tracking control: Theory and experiments. *International Journal of Control*, 74(14):1435 – 1446, 2001.

[11] M.H. Casado and F.J. Velasco. Surface ship trajectory control using non-linear backstepping design. *Proceedings of the Institute of Marine Engineering, Science and Technology Part A: Journal of Marine Engineering and Technology*, (3):3 – 8, 2003.

[12] Jasmin Velagic, Zoran Vukic, and Edin Omerdic. Adaptive fuzzy ship autopilot for track-keeping. *Control Engineering Practice*, 11(4):433 – 443, 2003.

[13] K.D. Do, Z.P. Jiang, and J. Pan. Underactuated ship global tracking under relaxed conditions. *IEEE Transactions on Automatic Control*, 47(9):1529 – 1536, 2002.

[14] Erjen Lefeber, Kristin Ytterstad Pettersen, and Henk Nijmeijer. Tracking control of an underactuated ship. *IEEE Transactions on Control Systems Technology*, 11(1):52–61, January 2003.

[15] Kristin Ytterstad Pettersen. Global uniform asymptotic stabilization of an underactuated surface vessel: Experimental results. *IEEE Transactions on Control Systems Technology*, 12(6):891–903, November 2004.

[16] Ker-Wei Yu and Chen-En Wu. Tracking control of a ship by PI-type sliding controller. *Journal of Marine Science and Technology*, 12(3):183 – 188, 2004.

[17] K.D. Do, Z.P. Jiang, and J. Pan. Global partial-state feedback and output-feedback tracking controllers for underactuated ships. *Systems and Control Letters*, 54(10):1015 – 1036, 2005.

[18] Khac Duc Do and Jie Pan. Global tracking control of underactuated ships with nonzero off-diagonal terms in their system matrices. *Automatica*, 41(1):87 – 95, 2005.

[19] A. Saeed, E.M. Attia, A.A. Helmy, and T. Awad. Design of a neuro-autopilot maneuvering controller for underactuated ships. *AEJ - Alexandria Engineering Journal*, 44(4):493 – 500, 2005.

[20] Jawhar Ghommam, Faical Mnif, Abderraouf Benali, and Nabil Derbel. Asymptotic backstepping stabilization of an underactuated surface vessel. *IEEE Transactions on Control Systems Technology*, 14(6):1150 – 1157, 2006.

[21] Witold Gierusz, Nguyen Cong Vinh, and Andrzej Rak. Maneuvering control and trajectory tracking of very large crude carrier. *Ocean Engineering*, 34(7):932 – 945, 2007.

[22] J. Cheng, J. Yi, and D. Zhao. Design of a sliding mode controller for trajectory tracking problem of marine vessels. *IET Control Theory and Applications*, 1(1):233 – 237, 2007.

[23] Hashem Ashrafiuon, Kenneth R. Muske, Lucas C. McNinch, and Reza A. Soltan. Sliding-mode tracking control of surface vessels. *IEEE Transactions on Industrial Electronics*, 55(11):4004 – 4012, 2008.

[24] L. Gyoungwoo, S. Surendran, and S.-H. Kim. Algorithms to control the moving ship during harbour entry. *Applied Mathematical Modelling*, 33(5):2474–2490, May 2009.

[25] Farbod Fahimi and Chris Van Kleeck. Alternative trajectory-tracking control approach for marine surface vessels with experimental verification. *accepted for publication in Robotica*, pages 1–12, February 2012.

[26] Y.-W. Huang and P.-C. Tung. Fuzzy pd system in adaptive control systems having input saturation. *Control and Intelligent Systems*, 35(3):217 – 222, 2007.

[27] Z. Li, J. Chen, G. Zhang, and M.G. Gan. Adaptive robust control for dc motors with input saturation. *IET Control Theory and Applications*, 5(16):1895 – 1905, 2011.

[28] Changyun Wen, Jing Zhou, Zhitao Liu, and Hongye Su. Robust adaptive control of uncertain nonlinear systems in the presence of input saturation and external disturbance. *IEEE Transactions on Automatic Control*, 56(7):1672 – 1678, 2011.

[29] Fabricio Garelli, Pablo Camocardi, and Ricardo J. Mantz. Variable structure strategy to avoid amplitude and rate saturation in pitch control of a wind turbine. *International Journal of Hydrogen Energy*, 35(11):5869 – 5875, 2010.

[30] Sangsoo Lim and Byoung Soo Kim. Aircraft cas design with input saturation using dynamic model inversion. *International Journal of Control, Automation and Systems*, 1(3):315 – 320, 2003.

[31] Javad Ahmadi, Ali Khaki Sedigh, and Mansour Kabganian. Adaptive vehicle lateral-plane motion control using optimal tire friction forces with saturation limits consideration. *IEEE Transactions on Vehicular Technology*, 58(8):4098 – 4107, 2009.

[32] Xiujian Yang and Zengcai Wang. Integrated control of braking and steering for vehicle stability considering the parametric uncertainties and input constraints. *International Journal of Vehicle Autonomous Systems*, 6(3-4):251 – 275, 2008.

[33] Mark W. Spong, James S. Thorp, and Jeffrey M. Kleinwaks. Control of robot manipulators with bounded input. *IEEE Transactions on Automatic Control*, AC-31(6):483 – 490, 1986.

[34] A. Zavala-Rio, E. Aguinaga-Ruiz, and V. Santibanez. Global trajectory tracking through output feedback for robot manipulators with bounded in puts. *Asian Journal of Control*, 13(3):430 – 438, 2011.

[35] Junfang Li and Tieshan Li. Design of ship's course autopilot with input saturation. *ICIC Express Letters*, 5(10):3779 – 3784, 2011.

[36] K.D. Do, Z.P. Jiang, and J. Pan. Universal controllers for stabilization and tracking of underactuated ships. *Systems and Control Letters*, 47(4):299 – 317, 2002.

[37] Chris Van Kleeck and Farbod Fahimi. Alternative trajectory-tracking control approach for marine surface vessels with experimental verification. *accepted for publication in Robotica*, pages 1–12, 2011.

[38] Yong-Yan Cao and Zongli Lin. Min-max mpc algorithm for lpv systems subject to input saturation. *IEE Proceedings: Control Theory and Applications*, 152(3):266 – 272, 2005.

[39] Alicia Arce, Alejandro J. Del Real, Carlos Bordons, and Daniel R. Ramirez. Real-time implementation of a constrained mpc for efficient airflow control in a pem fuel cell. *IEEE Transactions on Industrial Electronics*, 57(6):1892 – 1905, 2010.

[40] Y. Seki, J. Ohya, and M. Miyoshi. Collision avoidance system for vehicles applying model predictive control theory. *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC*, pages 453 – 458, 1999.

[41] Oyvind Hegrens, Jan Tommy Gravdahl, and Petter Tndel. Spacecraft attitude control using explicit model predictive control. *Automatica*, 41(12):2107 – 2114, 2005.

[42] Wenlin Wang and Daniel E. Rivera. Model predictive control for tactical decision-making in semiconductor manufacturing supply chain management. *IEEE Transactions on Control Systems Technology*, 16(5):841 – 855, 2008.

[43] S. Di Cairano, A. Bemporad, I.V. Kolmanovsky, and D. Hrovat. Model predictive control of magnetically actuated mass spring dampers for automotive applications. *International Journal of Control*, 80(11):1701 – 1716, 2007.

[44] Debangsu Bhattacharyya and Raghunathan Rengaswamy. System identification and nonlinear model predictive control of a solid oxide fuel cell. *Industrial and Engineering Chemistry Research*, 49(10):4800 – 4808, 2010.

[45] H. Eliasi, M.B. Menhaj, and H. Davilu. Robust nonlinear model predictive control for nuclear power plants in load following operations with bounded xenon oscillations. *Nuclear Engineering and Design*, 241(2):533 – 543, 2011.

[46] Gang Mei, Ahsan Kareem, and Jeffrey C. Kantor. Real-time model predictive control of structures under earthquakes. *Earthquake Engineering and Structural Dynamics*, 30(7):995 – 1019, 2001.

[47] Michael A. Henson. Nonlinear model predictive control: Current status and future directions. *Computers and Chemical Engineering*, 23(2):187 – 202, 1998.

[48] S. Joe Qin and Thomas A. Badgwell. An overview of nonlinear model predictive control applications. In *Nonlinear Predictive Control*, pages 369–392. Verlag, 2000.

[49] H.J. Kim, D.H. Shim, and S. Sastry. Nonlinear model predictive tracking control for rotorcraft-based unmanned aerial vehicles. In *American Control Conference, 2002. Proceedings of the 2002*, volume 5, pages 3576 – 3581 vol.5, 2002.

[50] H.J. Kim, D.H. Shim, and S. Sastry. Toward a robust model predictive controller applied to mobile vehicle trajectory tracking control. In *IFAC World Congress Milano (Italy)*, August-September 2011.

[51] Zhen Li, Jing Sun, and Soryeok Oh. Path following for marine surface vessels with rudder and roll constraints: An mpc approach. In *American Control Conference, 2009. ACC '09.*, pages 3611 –3616, june 2009.

[52] *http* : *//en.wikipedia.org/wiki/geodetic_system.*