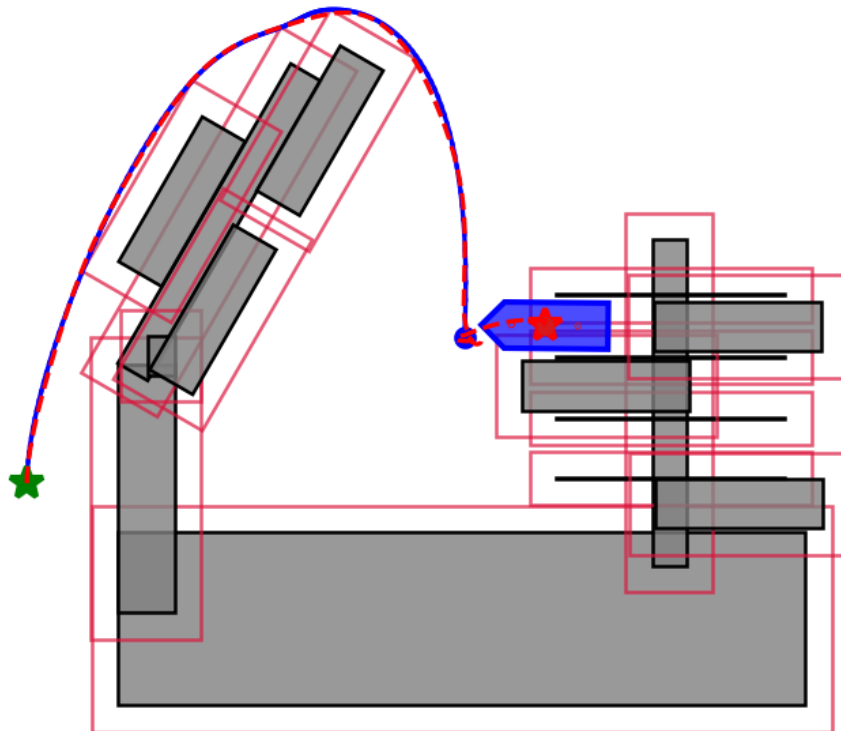




Autonomous Docking

System identification, Model Predictive Control and control allocation of marine vessels

FELIX PERSSON
OSKAR NILSSON

MASTER'S THESIS 2026

Autonomous Docking

System identification, Model Predictive Control and control
allocation of marine vessels

FELIX PERSSON
OSKAR NILSSON



Department of Mechanics and Maritime Sciences
Division of Vehicle Engineering and Autonomous Systems
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2026

Autonomous Docking
System identification, Model Predictive Control and control allocation of marine
vessels
FELIX PERSSON
OSKAR NILSSON

© FELIX PERSSON, OSKAR NILSSON 2026.

Supervisor: Daniel Söderberg, CPAC Systems AB
Examiner: Peter Forsberg, Department of Mechanics and Maritime Sciences

Master's Thesis 2026
Department of Mechanics and Maritime Sciences
Division of Vehicle Engineering and Autonomous Systems
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Cover: Docking maneuver in a simulation replica of harbour.

Typeset in L^AT_EX
Printed by Chalmers Reproservice
Gothenburg, Sweden 2026

Autonomous Docking

System identification, Model Predictive Control and control allocation of marine vessels

Felix Persson

Oskar Nilsson

Department of Mechanics and Maritime Sciences

Chalmers University of Technology

Abstract

This thesis investigates autonomous docking for monohull marine vessels, with the objective of developing a generalized maneuvering framework applicable across multiple monohull vessels. In today's application regarding autonomous docking, the solutions are often highly vessel-specific, which limits the ability to adapt.

To address this limitation, this thesis proposes a general maneuvering script capable of identifying different vessel dynamics and supporting the vessel in critical low-speed docking scenarios while maintaining safe and accurate behavior.

The proposed approach integrates vessel dynamic modeling together with advanced control and optimization techniques, including genetic algorithm, Model Predictive Control, and control allocation methods. A mathematical vessel model is used to approximate real-world behavior, which limits the controller and optimizer to realistic results. The genetic optimization process is used to tune the dynamic model based on real vessel behavior using recorded data, while Model Predictive Control ensures optimal decision-making over a finite horizon. The control allocator is applied to distribute control inputs among the available actuators in a feasible and efficient manner.

The results show that the proposed framework performs well in simulation environments and on a simulation rig, successfully executing autonomous docking maneuvers while adapting to various vessel characteristics. However, performance in real-world open-water experiments was limited due to model inaccuracies, hardware constraints, and time limitations, indicating that further development and testing are required for reliable real-world application.

Keywords: Autonomous docking, marine vessel identification, marine system modeling, model predictive control, control allocation.

Acknowledgements

This thesis was carried out at CPAC Systems AB during the spring of 2026.

We would like to thank CPAC Systems AB for providing the facilities and tools necessary for this thesis. We would also thank our supervisor at CPAC systems, Daniel Söderberg, for his continuous support, valuable advice, and genuine interest throughout the project. Furthermore, we would like to thank our Examiner at Chalmers University of Technology, Peter Forsberg, for his guidance and feedback during this period.

Finally, we would like to thank the people in the marine segment at CPAC Systems for providing the tools, information, and support that have helped us throughout this work.

Felix Persson & Oskar Nilsson, Gothenburg, June 2026

Thesis advisor: Daniel Söderberg, CPAC Systems AB

Thesis examiner: Peter Forsberg, Mechanics and Maritime Sciences

List of Acronyms

Below is the list of acronyms that have been used throughout this thesis listed in alphabetical order:

CAN	Controller Area Network
CB	Center of Buoyancy
CoG	Center of Gravity
CTE	Cross-Track Error
DOF	Degrees of Freedom
EKF	Extended Kalman Filter
EVC	Electronic Vessel Control
GA	Genetic Algorithm
GNSS	Global Navigation Satellite System
IMU	Inertial Measurement Unit
KF	Kalman Filter
MAM	Modified Abkowitz Model
MPC	Model Predictive Control
NED	North-East-Down
NMPC	Nonlinear Model Predictive Control
NLP	Nonlinear Programming
PANOC	Proximal Averaged Newton-type Optimization for Control
PID	Proportional-Integral-Derivative
PSO	Particle Swarm Optimization
RK4	Runge–Kutta 4
RMSE	Root Mean Square Error
RPM	Revolutions Per Minute
SI	System Identification
SOG	Speed Over Ground
USV	Unmanned Surface Vessel

Nomenclature

Below is the nomenclature of indices, sets, parameters, and variables used throughout this thesis.

Indices

i	Thruster index
k	Discrete time index
j	Cost function component index
N	Prediction horizon length

Sets

$\mathcal{T}$	Set of thrusters
$\mathcal{N}$	MPC prediction horizon indices $\{0, \dots, N - 1\}$
$\mathcal{S}$	State vector index set
$\mathcal{U}$	Control input index set

Parameters

Vessel and Geometry

m	Vessel mass
m_{ij}^{acc}	Effective mass (acceleration phase)
m_{ij}^{cst}	Effective mass (coast-down phase)
I_z	Yaw moment of inertia
x_g, y_g	Center of gravity coordinates
l_x, l_y	Thruster lever arms

Hydrodynamics

X_u, Y_v, N_r	Linear damping coefficients
X_{uu}, Y_{vv}, N_{rr}	Quadratic damping coefficients
m_{11}, m_{22}, m_{33}	Effective mass terms
X_0, Y_0, N_0	Constant bias forces and moment about CoG
Y_d	Differential thrust sway gain
N_d	Differential thrust yaw gain
X_H, Y_H, N_H	Hydrodynamic force and moment contributions
$X_{\text{ctrl}}, Y_{\text{ctrl}}, N_{\text{ctrl}}$	Control-generated forces and moment

Control and MPC

T_s	Sampling time
N_p	Prediction horizon
Q, R, Q_f	State, input, and terminal weighting matrices
Q_{CTE}	Cross-track error weight
Q_{acc}	Control input rate penalty weight
Q_{pos}, Q_{Ψ}	Terminal position and heading weights
Q_{Heading}	Heading error weight
λ_P, λ_R	Control allocation weights
M_{lat}	Meters per degree of latitude
Q_n	Weight on actuator power usage
Q_{δ}	Weight on thruster angle rate
Q_s	Weight on slack variables (force tracking error)
θ_{th}	Heading deadband threshold

Thruster Model

k_1, k_2	Propeller thrust coefficients
δ_{max}	Maximum thruster angle
$\dot{\delta}_{\text{max}}$	Maximum thruster slew rate
n_{max}	Maximum propeller speed (RPM)

Variables

States

x, y	Vessel position
ψ	Yaw angle
u, v	Surge and sway velocities
r	Yaw rate
u_{max}	Maximum surge velocity
v_{max}	Maximum sway velocity
r_{max}	Maximum yaw rate
X_{max}	Maximum surge force
Y_{max}	Maximum sway force
N_{max}	Maximum yaw moment
a_0^u, a_0^v	Initial surge and sway accelerations
$\dot{r}_0$	Initial yaw angular acceleration
d^u, d^v, d^r	Initial coast-down decelerations

Disturbances (EKF)

b_u, b_v, b_r	Estimated disturbance forces and moment
-----------------	---

Control Inputs

n_i	Propeller rotational speed (RPM)
-------	----------------------------------

δ_i	Thruster steering angle
T_i	Thruster force produced by actuator i
$\mathbf{u}$	Control input vector
$T_1^{\text{surge}}, T_1^{\text{sway}}, T_1^{\text{yaw}}$	Thruster force from step-response tests

Forces and Moments

X, Y, N	Surge force, sway force, and yaw moment
$\boldsymbol{\tau}$	Generalized force vector $[X \ Y \ N]^T$
$\boldsymbol{\tau}_d$	Desired generalized force vector
$\hat{\boldsymbol{\tau}}$	Allocated generalized force vector

Optimization

$\mathbf{s}$	Slack variables (unachievable portion of requested forces)
J	Optimization cost function

Navigation and Measurements

ϕ, λ	Latitude and longitude
ϕ_0, λ_0	Reference (initial) latitude and longitude
x_E, y_N	Position in East–North Cartesian frame
V_N, V_E	Velocity components in North–East frame
$u_{\text{body}}, v_{\text{body}}$	Body-fixed velocities derived from measurements
ψ_k, ψ_{k-1}	Heading at current and previous time step

Control Allocation

$\mathbf{B}(\boldsymbol{\delta})$	Actuator configuration matrix
$h(\cdot)$	Nonlinear mapping from actuator inputs to generalized forces

Contents

List of Acronyms	ix
Nomenclature	xi
List of Figures	xix
List of Tables	xxi
1 Introduction	1
1.1 Related work	1
1.1.1 Controller and trajectory tracking	2
1.1.2 Marine modeling and system identification	3
1.1.3 Purpose and goals	4
1.2 Research questions	4
1.3 Scope and limitations	5
2 Theory	7
2.1 Vessel maneuvering modeling	7
2.1.1 State space model with kinematics of a vessel	8
2.1.2 Fossen dynamic motion model	9
2.1.2.1 Coriolis and centripetal terms	9
2.1.2.2 Rigid-body inertia matrix	9
2.1.2.3 Linear damping matrix	10
2.1.3 Abkowitz dynamic motion model	10
2.1.3.1 Full Abkowitz polynomial expansion	11
2.1.3.2 Model reduction to linear hydrodynamic forces	12
2.1.4 Simplified modified Abkowitz model	12
2.1.4.1 Equations of motion implemented	13
2.1.4.2 Rigid-body and kinematic model	13
2.2 System identification	13
2.2.1 Overview of modeling and identification approaches	13
2.2.2 Genetic algorithm	14
2.2.3 Particle Swarm Optimization	14
2.2.4 Fossen modeling	14
2.2.5 Abkowitz modeling	14
2.2.6 Modified Abkowitz modeling	15
2.3 Path planning	15

2.3.1	A*	15
2.3.2	Interpolation	16
2.3.3	Path smoothing	16
2.4	Model Predictive Control	16
2.4.1	Dynamic models	17
2.4.2	Discretization	17
2.4.3	Nonlinear Model Predictive Control	18
2.4.4	Cost functions	18
2.4.4.1	Cross-track error	19
2.4.4.2	Control input/ acceleration	20
2.4.4.3	Terminal cost	20
2.4.5	Velocity cost	20
2.4.6	Heading cost	21
2.4.6.1	Obstacle and boundary avoidance	21
2.4.7	Optimization engine	22
2.5	Disturbance handling	22
2.5.1	Kalman Filter	22
2.5.2	Extended Kalman Filter	22
2.6	Control allocator	24
3	Methods	25
3.1	System identification	25
3.1.1	Step-response gray-box	25
3.1.2	Zigzag maneuver	28
3.1.3	Drag constant identification surge	28
3.1.4	Hyperparameter identification	29
3.1.5	Implementing the vessel model	30
3.1.6	Genetic Algorithm-based parameter estimation	31
3.2	Model Predictive Control	32
3.2.1	Prediction model	32
3.2.2	Cost function	32
3.2.3	Solver	32
3.3	Control allocation	33
3.3.1	General control allocation formulation	35
3.4	CAN bus communication	37
3.4.1	Transmit TX	38
3.4.2	Receive Rx	38
4	Results	41
4.1	Model validation	41
4.1.1	Calibration on simulation rig	41
4.2	Validation of MPC	48
4.3	Validation of Extended Kalman Filter	50
4.3.1	Acceleration vs IMU	55
4.4	Comparison of the complete system	56
4.4.1	Scenario 1 on simulation rig	57
4.4.2	Scenario 2 on simulation rig	58

4.4.3	Scenario 3 on simulation rig	60
4.4.4	Extended Kalman filter compensation	65
4.5	The entire system real-world test on open water	67
4.5.1	Calibration first data collection	68
4.5.2	Calibration second data collection	72
4.5.3	Real model predictive controller tests	78
4.6	Control allocation	80
4.6.1	Twin thrusters at the stern in simulation rig	80
5	Discussion	83
5.1	Dynamic model	83
5.2	Model predictive controller	83
5.3	Extended Kalman Filter	84
5.4	Control allocation	85
5.5	Whole system	85
5.5.1	Simulation	85
5.5.2	Real tests	86
5.6	Future work	86
6	Conclusion	89
	Bibliography	91

List of Figures

2.1	3-DOF overview of a marine vessel.	7
2.2	Abkowitz model of the ship motion coordinate system.	11
2.3	CTE illustration.	19
2.4	EKF inside a control loop.	23
2.5	Control system structure when control allocation is used.	24
3.1	Calibration Step-response, on the dummy simulation station.	27
3.2	20/20 zigzag trajectory.	28
3.3	Two thrusters, T_P as the thrust from port thruster and T_S as the thrust from starboard thruster. Their location on the marine vessel given with the lengths l_Y and l_x	34
3.4	A simple illustration of a simple CAN bus.	37
4.1	Simulation rig setup.	42
4.2	Logged step response conducted on the simulation rig.	42
4.3	2020-zigzag calibration plot of important states.	44
4.4	The position tracking from the GA-Algorithm	45
4.5	The state tracking from the GA-Algorithm.	46
4.6	RMSE of important GA identified states.	47
4.7	MPC controller architecture illustration.	48
4.8	MPC validation using python models as plant.	49
4.9	States of MPC validation using python models as plant.	49
4.10	MPC validation using python models as plant.	50
4.11	States of MPC validation using python models as plant.	50
4.12	Plot of a static EKF Simulation	51
4.13	Plot of a static EKF Simulation, disturbance estimates.	52
4.14	Plot of EKF Simulation, constant surge force.	53
4.15	Plot of EKF Simulation, no added bias.	54
4.16	Plot of EKF Simulation, 500 N added bias in surge	54
4.17	Disturbance estimate plot of EKF Simulation, 500 N added bias in surge.	55
4.18	Acceleration derived from GPS-velocity vs IMU data.	56
4.19	Velocity plot for two coast-down maneuvers.	56
4.20	Simple trajectory following.	57
4.21	Plots of CTE, heading error and the corresponding control inputs. . .	58
4.22	Trajectory following.	59
4.23	Plots of CTE, heading error and the corresponding control inputs. . .	60

4.24	Scenario 3: Obstacle avoidance with tight docking.	61
4.25	Plots of CTE, heading error and the corresponding control inputs. . .	61
4.26	Scenario 3.2: Obstacle avoidance with tight docking.	62
4.27	Plots of CTE, heading error and the corresponding control inputs. . .	63
4.28	Scenario 3.3: Obstacle avoidance with tight docking.	64
4.29	Plots of CTE, heading error and the corresponding control inputs. . .	64
4.30	Disturbance handling trajectory.	65
4.31	Plots of CTE, heading error and the corresponding control inputs for disturbance handling.	66
4.32	EKF bias estimate for constant disturbance.	67
4.33	Targa 46 boat used in the open water validation.	67
4.34	Real-World step-response including rpm-sweep.	68
4.35	Zigzag calibration from rel life tests.	70
4.36	Simulation vs measured position.	70
4.37	Real vs identified states.	71
4.38	RMSE for position and heading between the simulated and measured trajectory.	72
4.39	Step-response calibration from real vessel.	73
4.40	Zigzag calibration 3.2.	75
4.41	Real measured position track vs simulated position track.	76
4.42	Measured vs simulated states from GA identification.	77
4.43	RMSE for position and heading between the simulated and measured trajectory.	77
4.44	Trajectory obtained during the full system validation.	78
4.45	Solver time during the real system test.	78
4.46	Computed control inputs during the test.	79
4.47	Cross-track error (CTE) during the experiment.	79
4.48	Heading error during the experiment.	79
4.49	The resulting path of the allocator outputs to the marine vessel. . . .	80
4.50	Result of the allocator forces provided after request from MPC with same force limits.	81
4.51	Result of the allocator commands with angle δ and RPM of both thrusters.	81

List of Tables

2.1	State, velocity, and force vectors for 6-DOF and 3-DOF vessel models.	8
3.1	Thruster positions in the body-fixed coordinate frame.	33
3.2	CAN bus received signals from the vessel's navigation system.	39
4.1	Parameters derived from vessel step-response calibration (updated). .	43
4.2	Identified Abkowitz maneuvering model parameters (GA-updated). .	46
4.3	Weighting factors used in GA-based parameter identification.	48
4.4	Parameters derived from step-response calibration (updated vessel test).	69
4.5	Identified Abkowitz maneuvering model parameters (GA fit with pinned calibration parameters).	71
4.6	Parameters derived from step-response calibration (updated vessel test).	74
4.7	Identified Abkowitz maneuvering model parameters (GA with equilibrium-consistent bounds).	76

1

Introduction

In recent years, the demand for precise and reliable digital anchoring and autonomous docking systems has grown significantly. This development is driven by advances in autonomous vehicles, increasingly capable actuation systems, and the emergence of more powerful onboard computing platforms. Traditionally, high-precision dynamic positioning has been limited to purpose-built commercial installations. However, this has started to change as sensors have become cheaper and steer-by-wire systems have been adopted in the leisure segment. Leisure and smaller commercial vessels typically lack advanced dynamic models and builders do not have the expertise to tune complex control systems or models.

The aim of this master thesis is to address the gap in current approaches to autonomous docking by developing a general model-based controller that can work on different leisure vessels and achieve higher precision than today's simple controllers in autonomous docking systems. In addition, the thesis will investigate how well a control allocator will perform and compare its effectiveness with alternative control strategies.

The approach is broad and general. By collecting relevant system data and applying concepts from system identification, Model Predictive Control (MPC), nonlinear filtering, the limitations of the current systems can be reduced. The study seeks to contribute to the development of an autonomous docking system that is adaptable, robust, and capable of operating reliably on a wide range of monohull vessels with minimum advanced tuning.

1.1 Related work

The docking of marine vessels is a challenging maneuver that demands precise control, accurate modeling of complex vessel dynamics, and the ability to withstand harsh and unpredictable environmental disturbances. Traditional rule-based docking approaches often struggle with generalization and lack robustness when applied to highly dynamic and uncertain maritime scenarios [1].

This section provides a brief description of how different key terms, concepts and techniques are used in this thesis. The material presented here builds on previous research and serves as an initial gathering of information on the subject. The thesis aims to extend and improve an existing research project conducted in collaboration with CPAC Systems and the authors of [2].

1.1.1 Controller and trajectory tracking

Path following for a marine vessel forms the foundation of an autonomous docking vessel. This process can be divided into two key components. Generating a trajectory based on desired waypoints and designing a control system capable of steering the vessel along that trajectory. Since this task is intended to be fully autonomous, a robust control framework is required to track the reference trajectory under varying operating conditions.

In earlier studies on Unmanned Surface Vessels (USVs) conducted by [3], a Proportional-Integral-Derivative (PID) controller was proposed to mitigate external disturbances and regulate vessel movements. Their system was implemented and validated on a cyber-physical test platform, demonstrating promising performance for surge and heading control. However, the results show a significant lateral tracking error. The study concluded that under dynamic conditions and higher-magnitude disturbances, the PID-based approach was insufficient and more advanced control strategies would be required to ensure reliable path following.

A related study investigating the use of a nonlinear PID controller used for the control of a large container ship under the influence of wind and currents was presented by [4]. In this work, the reference path was generated offline with predefined surge speeds assigned to each segment, without explicit consideration of the underlying dynamics of the vessels. The proposed approach showed promising results in a simulation environment, where the integral action managed to reduce the steady-state errors. The authors noted that the PID controller requires carefully performed system identification to achieve good performance in real sea trials and reduce modeling errors. To address these limitations, the study pointed toward reinforcement-learning-based MPC as a potential improvement, referencing the work of [5]. Such methods reduce dependence on highly accurate vessel models and lessen the need for fine-tuning of control parameters. Both studies share a key limitation: they do not account for obstacles or dynamically moving objects that may interfere with the planned trajectory. Their primary objective is to track a predefined path, rather than handle the broader challenges associated with autonomous docking in dynamic environments. A key takeaway from the PID approach is that it is a reactive controller acting on errors. As shown in [6], the MPC outperformed the optimized PID in many ways.

The study presented in [2] addresses trajectory planning and dynamic route adaptation for autonomous marine vessel docking using MPC. The authors demonstrate that their framework can generate feasible and collision-free trajectories in both static and dynamic environments. The subsequent control implementation relies on a linear, purely kinematic MPC formulation, which neglects essential nonlinear and hydrodynamic characteristics of marine vessels. As a result, several important real-world effects such as masses, dampening and disturbances were not represented or handled. Moreover, the system did not include filtering or estimation of measurements, leaving it unable to handle noise making the closed loop control system overly sensitive. To enhance the reliability and realism of the autonomous docking system, the authors recommend integrating a more comprehensive dynamic vessel model that captures external forces from wind, currents, and varying water conditions. In addition, they emphasize the importance of extensive testing in complex

real-world scenarios.

Previous studies investigating MPC for autonomous docking have concluded that the system shows potential to automate the docking procedure, while ensuring safety and efficiency by combining path planning algorithms with MPC. The ability to safely react to static and dynamic obstacles showed promising results [2].

1.1.2 Marine modeling and system identification

Modeling a marine vessel poses several challenges, particularly in obtaining accurate and reliable parameter identification. This step is crucial to obtain the vessel model that ensures a consistent and controlled behavior. As described by Fossen [7], marine surface vessels are commonly modeled using a 3-DOF representation, capturing the generalized position and velocity in the horizontal plane. This formulation allows external forces and moments to be incorporated using matrix-vector expressions. Hydrodynamic effects such as mass, added mass, and linear and nonlinear damping are typically decomposed into contributions along the surge, sway, and yaw axes. The added-mass Coriolis and centripetal components can be combined into a single matrix representation, which avoids unnecessary re-parameterization and keeps the model computationally efficient. The objective is to obtain a sufficiently simple and controllable model that still captures the relevant physical dynamics.

Analyzing the motion of a marine vessel, it is convenient to define it as two Earth-coordinate frames, the Earth-centered reference frame and the body-fixed reference frame [7]. The inertial frame is typically expressed in the *North-East-Down* (NED) coordinate system, indicated in Equation 1.1.

$$n = (x_n, y_n, z_n), \quad (1.1)$$

The x_n axis points toward true north, y_n points towards the true east, and z_n points down into the surface seen in 1.1. The body-fixed reference frame (BODY) is described as Equation 1.2.

$$b = (x_b, y_b, z_b), \quad (1.2)$$

The body frame in Equation 1.2 is attached to the vessel and describes its position and orientation relative to the NED frame. The origin of the body-fixed frame must remain fixed on the vessel, and the choice of location depends on the modeling objectives. In this thesis, the body-fixed origin is placed at the assumed center of buoyancy (CB).

$$r_{bb}^b = [x_b, y_b, z_b]^\top. \quad (1.3)$$

Fossen [7] describes the transformation between the NED and body-fixed frames using a rotation matrix $\mathbf{R}$. For 3-DOF surface vessel $\mathbf{R} \in \mathbb{R}^{3 \times 3}$ maps body-fixed velocities to earth-fixed velocities in the horizontal plane. In the general case, this matrix is constructed from the principal rotations of the roll, pitch, and yaw axes. However, since only rotations in yaw are considered in this thesis, the rotation matrix reduces to a purely geometric yaw transformation, as shown in Equation 1.4.

$$\mathbf{R}(\psi) = \begin{bmatrix} \cos(\psi) & -\sin(\psi) & 0 \\ \sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (1.4)$$

This simplified type of vessel modeling is widely used in research [8] [9] [10], as many studies on autonomous docking focus exclusively on motion in the horizontal plane. Under such conditions, a reduced 3-DOF is sufficient to capture the dominant dynamics influencing maneuvering dynamics. The paper "Nonlinear Low-level Controller Design for an Autonomous Surface Vehicle" [9], demonstrates that the Fossen dynamic implementation showed reasonable results that reflect the expectations of the behavior of a real vessel.

When maneuvering a marine vessel, multiple parameters are included that affect the model and the behavior of the vessel. To be able to find solutions for the mass and hydrodynamic parameters, some sort of identification is needed. Since the thesis aims to create a model that works across multiple vessels, a general model and algorithm are necessary. Using a genetic algorithm, more specifically a Distributed evolutionary algorithms in python (DEAP) [11]. It is possible to use the algorithm to try many different unknown parameters and try them to see the difference and error, with the help of root mean square error (RMSE).

1.1.3 Purpose and goals

The purpose of this thesis is to investigate the application of MPC towards a dynamic vessel model. More specific, it investigates whether a MPC can be applied to control the path of a leisure vessel in a docking scenario. Additional to this, to see if a general model can be applied and constructed to be used for multiple leisure vessels.

1.2 Research questions

The thesis aims to develop an autonomous docking system tailored for marine monohull vessels operating under real-world conditions. The work focuses on vessels in the 20-100 feet range and aims to advance the modeling, identification, and control strategies required for reliable autonomous docking. To achieve this, the project is structured around several key research questions.

- How can a generalized dynamic model be developed to handle monohull vessels in the range 20-100 feet?
- What type of calibration procedure is most suitable to identify the parameters of the dynamic model for monohull vessels?
- Which system identification model structure provides the best representation of monohull vessels dynamic behavior during docking maneuvers?
- What type of measurement data is needed for an accurate system identification?
- Which sensor fusion approach is the most suitable to provide the most robust state estimation?
- Which model based control structure is suitable for an autonomous docking procedure?
- What is a suitable control allocation model for autonomous docking?

1.3 Scope and limitations

This thesis builds upon previous work conducted in the field of autonomous docking where several methodological and practical limitations are present. The proposed algorithms and experimental evaluations are constrained by the available hardware, software frameworks, and test environments. In particular, the scope of this work is limited to systems compatible with Volvo Penta propulsion solutions and CPAC Systems control systems, as well as the availability of marine vessels at Krossholmen harbour.

All experimental and validation scenarios are conducted in controlled environments, where dynamic obstacles and external objects can be regulated. This allows for systematic testing and repeatability, but limits the ability to fully capture the complexity of unstructured real-world harbour environments. The vessel dynamics are represented by a generalized dynamic model intended to fit multiple monohulls with different characteristics. To reduce computability and controllability the vessel motion is restricted to three degrees of freedom, surge, sway and yaw. The focus is to maneuver a marine monohull vessel in a horizontal plane, the need for more DOF is neglected. Testing and validating the dynamics and proposed control strategy will be simulated in a python environment, a Unity environment, and finally in real world scenarios.

The execution of this thesis is divided into several stages. The first stage focuses on developing a dynamic marine vessel model capable of accounting for environmental disturbances. This model is subsequently validated using real logged measurement data. The next stage aims to generalize the modeling approach so that it can be applied to multiple marine vessels, while retaining the ability to calibrate and adapt the model to a specific vessel. In the final stage, the developed model is integrated into a control framework designed to ensure robust performance in the presence of disturbances. The complete modeling, estimation, and control pipeline is intended to be tested and validated on real marine vessels, subject to practical and time-related constraints.

2

Theory

This chapter establishes the theoretical foundation of the study. It describes the dynamics of marine vessels, introduces key principles of system identification, and outlines the necessary background on MPC, Kalman Filter, and control allocation.

2.1 Vessel maneuvering modeling

The movement of a vessel in space can be described using six-degree-of-freedom equations when analyzing vessel maneuvering [7]. These equations describe both the dynamics of the vessel and the hydrodynamics that affect the system, see Table 2.1. In this section, different ways of describing these dynamics are investigated and presented to find a reasonable model suitable for the goal of this thesis. For simplicity, this thesis considers a reduced state-space model with 3-DOF, referring to surge (longitudinal motion), sway (sideways motion), and yaw (heading rotation), see Figure 2.1.

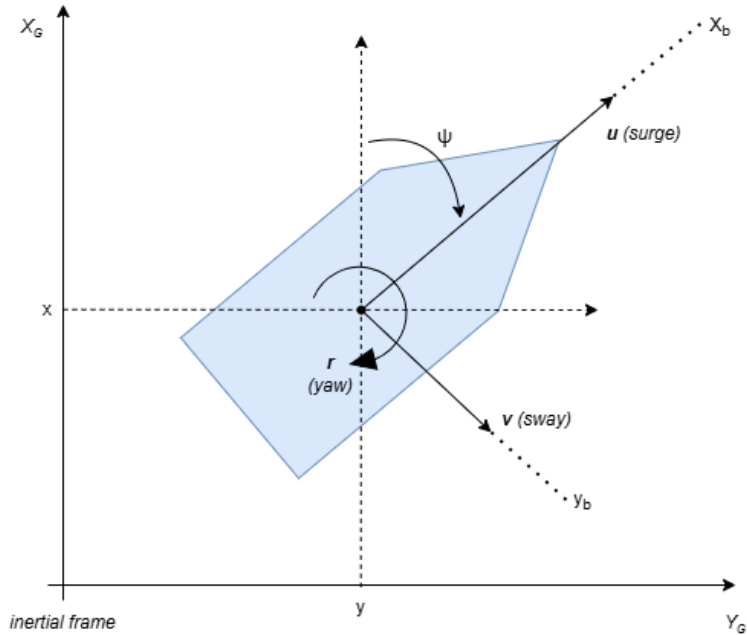


Figure 2.1: 3-DOF overview of a marine vessel.

Table 2.1: State, velocity, and force vectors for 6-DOF and 3-DOF vessel models.

DOF	Parameters	Description
6	$\eta = [x \ y \ z \ \phi \ \theta \ \psi]^T \in \mathbb{R}^6$	Orientation of the vessel given by position and Euler angles in an inertial body frame.
	$\nu = [u \ v \ w \ p \ q \ r]^T \in \mathbb{R}^6$	Linear and angular velocities in a body-fixed coordinate system.
	$\tau = [X \ Y \ Z \ K \ M \ N]^T \in \mathbb{R}^6$	Forces and moments in a body-fixed coordinate system.
3	$\eta = [x \ y \ \psi]^T \in \mathbb{R}^3$	Orientation in the horizontal plane given by Cartesian position (x, y) and heading angle ψ .
	$\nu = [u \ v \ r]^T \in \mathbb{R}^3$	Linear and angular velocities in a body-fixed coordinate system in surge (u), sway (v) and yaw (r).
	$\tau = [X \ Y \ N]^T \in \mathbb{R}^3$	Forces and moments in a body-fixed coordinate system in surge (X), sway (Y) and yaw (N).

2.1.1 State space model with kinematics of a vessel

The maneuvering of a ships dynamics can be expressed using a state-space model,

$$\dot{x} = f(x, c) + q(t). \quad (2.1)$$

Where x is the state vector, c is the control input vector, expressed through the transition function $f(x, c)$ [8], with $q(t)$ denoting the processed noise. For surface vessels, the generalized position vector with three degrees of freedom (3-DOF) can be expressed as $\eta = [x, y, \psi]^T$ and the velocity vector $\nu = [u, v, r]^T$. The full state vector resulting in

$$x = \begin{bmatrix} \eta \\ \nu \end{bmatrix} = \begin{bmatrix} x_0 \\ y_0 \\ \psi \\ u \\ v \\ r \end{bmatrix}. \quad (2.2)$$

Here, x_0 and y_0 denote the Earth-fixed position coordinates, while ψ describes the heading angles. The velocities represent the surge velocity u , sway velocity v , and yaw velocity r . The Kinematic position model can be expressed as:

$$\dot{\eta} = \mathbf{R}(\psi)v, \quad (2.3)$$

where $\mathbf{R}(\psi)$ is the given rotation matrix,

$$\mathbf{R}(\psi) = \begin{bmatrix} \cos(\psi) & -\sin(\psi) & 0 \\ \sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (2.4)$$

2.1.2 Fossen dynamic motion model

The maneuvering model for the marine vessel, as formulated by Fossen can be expressed as shown in Equation 2.5. Linear and nonlinear maneuvering equations are expressed to capture the behavior of the vessel.

$$\underbrace{M_{\text{RB}}\dot{v} + C_{\text{RB}}(v)v}_{\text{rigid-body forces}} + \underbrace{M_A\dot{v}_r + N(v_r)v_r}_{\text{hydrodynamic forces}} = \tau + \tau_{\text{wind}} + \tau_{\text{wave}} \quad (2.5)$$

2.1.2.1 Coriolis and centripetal terms

The term $N(v_r)$ accounts for hydrodynamic damping, while the added mass and Coriolis/centripetal effects are collected in the inertia matrix [7]. The velocity related to the water is defined as $v_r = v - v_c$, where $v_c = [u_c, v_c, 0]^T$ is the current velocity expressed in the body frame coordinates. The body-fixed acceleration vector, $\dot{v}$, is expressed as:

$$\dot{v} = \begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{r} \end{bmatrix}, \quad (2.6)$$

where $\dot{u}$, $\dot{v}$, and $\dot{r}$ denotes surge acceleration, sway acceleration, and yaw angular acceleration respectively. The hydrodynamic matrix $N(v_r)$ is defined as:

$$N(v_r) := C_A(v_r) + D + D_n(v_r), \quad (2.7)$$

where C_A is the added-mass Coriolis matrix, D is the linear damping matrix, and $D_n(v_r)$ contains nonlinear damping terms. The reason for this addition of parameters is to avoid over parameterization and maintaining a physically meaningful model structure.

2.1.2.2 Rigid-body inertia matrix

The rigid-body inertia matrix M_{RB} , represents the mass and rotational inertia of the vessel, where m is the mass of the vessel and I_z is the yaw moment of the inertia about the vertical axis.

$$M_{\text{RB}} = \begin{bmatrix} m & 0 & 0 \\ 0 & m & mx_g \\ 0 & mx_g & I_z \end{bmatrix}. \quad (2.8)$$

Here x_g represents the center-of-gravity in the longitudinal position in the body-fixed frame. The added-mass matrix M_A represents the inertia of the fluid surrounding the hull, which resists changes in motion.

$$M_A = - \begin{bmatrix} X_{\dot{u}} & 0 & 0 \\ 0 & Y_{\dot{v}} & Y_{\dot{r}} \\ 0 & N_{\dot{v}} & N_{\dot{r}} \end{bmatrix}, \quad (2.9)$$

where $X_{\dot{u}}$, $Y_{\dot{v}}$, $Y_{\dot{r}}$, $N_{\dot{v}}$, $N_{\dot{r}}$ represents the hydrodynamic parameters of the surrounding fluids.

The rigid-body Coriolis and centripetal matrix $C_{RB}(\nu)$ represents the rotation body-fixed frame that couples the surge, sway, and yaw dynamics ensuring that no energy is added or removed. The added-mass Coriolis matrix $C_A(\nu_r)$ is related to $C_{RB}(\nu)$ and defined based on the velocity relative to the water, ν_r .

The rigid-body Coriolis matrix is given by:

$$C_{RB}(\nu) = \begin{bmatrix} 0 & -mr & -mx_g r \\ mr & 0 & 0 \\ mx_g r & 0 & 0 \end{bmatrix}. \quad (2.10)$$

The added-mass Coriolis and centripetal matrix is expressed as

$$C_A(\nu_r) = \begin{bmatrix} 0 & 0 & Y_{\dot{v}}v_r + Y_{\dot{r}}r \\ 0 & 0 & -X_{\dot{u}}u_r \\ -Y_{\dot{v}}v_r - Y_{\dot{r}}r & X_{\dot{u}}u_r & 0 \end{bmatrix}. \quad (2.11)$$

2.1.2.3 Linear damping matrix

The linear damping matrix D , represents the low-speed hydrodynamic drag. Where the forces are approximately proportional to velocity:

$$D = - \begin{bmatrix} X_u & 0 & 0 \\ 0 & Y_v & Y_r \\ 0 & N_v & N_r \end{bmatrix}. \quad (2.12)$$

The matrix $D_n(\nu_r)$ represent the nonlinear damping, which should be active if there is a quadratic and higher-order drag, which is important at higher speeds. If the operation speeds are low or a simplified model is desired, a common assumption is to approximate it to zero:

$$D_n(\nu_r) \approx 0. \quad (2.13)$$

As described in [7] the ocean current with constant magnitude V_c and direction β_{V_c} generates a body-fixed current velocity vector $v_c = [u_c, v_c, 0]^T$. It can be computed as:

$$u_c = V_c \cos(\beta_{V_c} - \psi), \quad (2.14)$$

$$v_c = V_c \sin(\beta_{V_c} - \psi). \quad (2.15)$$

The corresponding current acceleration in the body frame is given by:

$$\dot{u}_c = r V_c \sin(\beta_{V_c} - \psi) = r v_c, \quad (2.16)$$

$$\dot{v}_c = -r V_c \cos(\beta_{V_c} - \psi) = -r u_c. \quad (2.17)$$

2.1.3 Abkowitz dynamic motion model

A widely used model to describe the vessel's motion in 3-DOF is the Abkowitz model, which approximates the hydrodynamic forces acting on the vessel using Taylor-series expansion around a steady operating point. The overview of the coordinate system can be seen in Figure 2.2.

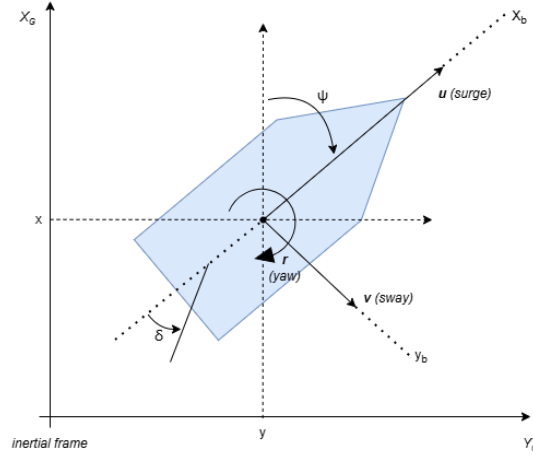


Figure 2.2: Abkowitz model of the ship motion coordinate system.

The general idea of Abkowitz model is to approximate the hydrodynamic forces through a polynomial expansion of the surge velocity u , sway velocity v , yaw rate r , and rudder angle δ [12]. The forces can then be expressed as the following:

$$X = f_1(u, v, r, \delta), \quad (2.18)$$

$$Y = f_2(u, v, r, \delta), \quad (2.19)$$

$$N = f_3(u, v, r, \delta), \quad (2.20)$$

where u is the velocity in surge, v is the velocity in sway, r is the yaw rate, and δ is the rudder angle.

2.1.3.1 Full Abkowitz polynomial expansion

The functions f_1, f_2 , and f_3 contains a sum of linear and nonlinear hydrodynamic derivatives. The surge polynomial expansion:

$$\begin{aligned} f_1(u, v, r, \delta) = & X_1 + X_u u + X_{uu} u^2 + X_{uuu} u^3 + X_{vv} v^2 + (X_{rr} + m x_G) r^2 \\ & + X_{\delta\delta} \delta^2 + (X_{vr} + m) v r + X_{v\delta} v \delta + X_{r\delta} r \delta \\ & + X_{vvu} v^2 u + X_{rru} r^2 u + X_{\delta\delta u} \delta^2 u \\ & + X_{vru} v r u + X_{v\delta u} v \delta u + X_{r\delta u} r \delta u. \end{aligned} \quad (2.21)$$

The sway polynomial expansion:

$$\begin{aligned} f_2(u, v, r, \delta) = & Y_1 + Y_u u + Y_{uu} u^2 + Y_{uuu} u^3 + Y_{vv} v + Y_{rr} r + m r u + Y_{\delta\delta} \delta \\ & + Y_{vvv} v^3 + Y_{rrr} r^3 + Y_{\delta\delta\delta} \delta^3 \\ & + Y_{vvr} v^2 r + Y_{vv\delta} v^2 \delta + Y_{vrr} v r^2 + Y_{\delta rr} \delta r^2 \\ & + Y_{v\delta\delta} v \delta^2 + Y_{r\delta\delta} r \delta^2 + Y_{vr\delta} v r \delta \\ & + Y_{vu} v u + Y_{ru} r u + Y_{\delta u} \delta u \\ & + Y_{vuu} v u^2 + Y_{ruu} r u^2 + Y_{\delta uu} \delta u^2. \end{aligned} \quad (2.22)$$

The yaw moment expansion:

$$\begin{aligned}
 f_3(u, v, r, \delta) = & N_1 + N_u u + N_{uu} u^2 + N_{uuu} u^3 + N_{vv} v + N_{rr} r + N_{\delta\delta} \delta \\
 & + N_{vvv} v^3 + N_{rrr} r^3 + N_{\delta\delta\delta} \delta^3 \\
 & + N_{vvr} v^2 r + N_{vv\delta} v^2 \delta + N_{vrr} v r^2 + N_{\delta rr} \delta r^2 \\
 & + N_{v\delta\delta} v \delta^2 + N_{r\delta\delta} r \delta^2 + N_{vr\delta} v r \delta \\
 & + N_{vu} v u + N_{ru} r u - m x_G r u + N_{\delta u} \delta u \\
 & + N_{vu^2} v u^2 + N_{ru^2} r u^2 + N_{\delta u^2} \delta u^2.
 \end{aligned} \tag{2.23}$$

2.1.3.2 Model reduction to linear hydrodynamic forces

Because these equations involve a large number of hydrodynamic derivatives, parameter identification becomes challenging. The many cross-correlated parameters reduce the robustness of the estimation process and increase its sensitivity. A common approach is therefore to restrict the model to linear hydrodynamic forces only. The functions can then be expressed as:

$$f_1(u, v, r, \delta) = X_1 + X_u u + X_v v + X_r r + X_\delta \delta, \tag{2.24}$$

$$f_2(u, v, r, \delta) = Y_1 + Y_u u + Y_v v + Y_r r + Y_\delta \delta, \tag{2.25}$$

$$f_3(u, v, r, \delta) = N_1 + N_u u + N_v v + N_r r + N_\delta \delta. \tag{2.26}$$

Linearization lowers the model's complexity while retaining the essential hydrodynamic characteristics required for system identification and controller design.

2.1.4 Simplified modified Abkowitz model

The simplified modified Abkowitz model enables the reduction in the number of hydrodynamic parameters. This reduction is motivated by the need to prevent over-parameterization and to enable data-driven parameter estimation.

Several recent studies emphasize the need to reduce the classical maneuvering model such as Abkowitz due to multicollinearity and parameter identification problems when using real test data. As Alexandersson shows in his study [8], systematically reducing the maneuvering model and reducing the order of the model would lead to a more reliable identification and improve robustness.

To address this, the simplified modified Abkowitz model:

- removes all added-mass derivatives ($X_{\dot{u}}$, $Y_{\dot{v}}$, $N_{\dot{r}}$) and their cross-coupled terms ($Y_{\dot{r}}$, $N_{\dot{v}}$);
- keeps only the rigid-body Coriolis terms (mvr , $-mur$, $m x_G r^2$) explicitly;
- retains a reduced set of nonlinear hydrodynamic terms through a truncated Taylor expansion (e.g. $X_{uu}|u|u$, $Y_{vv}|v|v$, $N_{rr}|r|r$);
- includes mixed nonlinear terms (e.g. X_{vr} , Y_{vvr} , N_{vrr}) when they proved beneficial for model fidelity;
- uses externally applied forces and moments [X_{ctrl} , Y_{ctrl} , N_{ctrl}] to incorporate rudder and propulsion effects.

2.1.4.1 Equations of motion implemented

The resulting dynamic equations follow the classical Abkowitz structure, but with a significantly reduced parameter set:

$$m_{11}\dot{u} = mvr + mx_g r^2 + X_H + X_{\text{ctrl}} + X_0, \quad (2.27)$$

$$m_{22}\dot{v} + m_{23}\dot{r} = -mur + Y_H + Y_{\text{ctrl}} + Y_0, \quad (2.28)$$

$$m_{32}\dot{v} + m_{33}\dot{r} = -mx_g ur + N_H + N_{\text{ctrl}} + N_0, \quad (2.29)$$

where the hydrodynamic forces X_H , Y_H , and N_H are computed from a truncated and simplified Taylor-series expansion. For example:

$$X_H = X_u u + X_{uu}|u|u + X_{vv}v^2 + X_{rr}r^2 + X_{vr}vr, \quad (2.30)$$

$$Y_H = Y_v v + Y_{vv}|v|v + Y_{rr}r^2 + Y_{uv}uv + Y_{ur}ur + Y_{vr}vr, \quad (2.31)$$

$$N_H = N_r r + N_{rr}|r|r + N_{vv}v^2 + N_{vr}vr + N_{uv}uv + N_{ur}ur. \quad (2.32)$$

The parameters X_0 , Y_0 , and N_0 are the constant hydrodynamic forces (bias) and moment. They help the model match real data. Analogous expressions for sway and yaw capture the dominant nonlinear velocity terms.

2.1.4.2 Rigid-body and kinematic model

The rigid-body Coriolis terms are preserved explicitly because they represent essential physical coupling:

$$\text{RB}_{\text{surge}} = mvr + mx_g r^2, \quad \text{RB}_{\text{sway}} = -mur, \quad \text{RB}_{\text{yaw}} = -mx_g ur. \quad (2.33)$$

These terms capture the cross-coupling between surge, sway, and yaw dynamics and are therefore present even if other hydrodynamic derivatives are removed. Meanwhile, Earth-fixed kinematics follows the standard coordinate transformation, which can be seen in Equation 2.4.

2.2 System identification

This section provides an overview of established approaches to system modeling and system identification for marine vessels. The presentation is based on previous studies and evaluations reported in [10]. The focus is on methods suitable for low-speed maneuvering and data-driven identification, which are particularly relevant for autonomous marine vessels operating in confined and dynamic environments.

2.2.1 Overview of modeling and identification approaches

Marine vessel modeling and identification approaches can generally be categorized into physics-based and data-driven methods. Physics-based models are derived from first principles, such as Newton-Euler equations, and represent hydrodynamic effects through parameterized force and moment expressions obtained, for example from Taylor-series expansions.

The optimization-based identification methods rely on experimental or operation data to estimate the parameters in the model by minimizing the difference between measured and simulated marine vessel behavior. Such approaches are well suited for low-speed maneuvering where information regarding the hydrodynamics in the marine vessel model can be identified.

2.2.2 Genetic algorithm

Genetic algorithms (GAs) are stochastic optimization methods inspired by principles of natural evolution. Given a parameterized model, constraints, and a defined cost function, GAs search for parameter sets that minimize the model error [11]. In system identification, candidate parameter sets are evaluated by simulating the vessel model on specific measured data and compare the results using root mean square error (RMSE). Through selection, crossover, and mutation, the population evolves toward solutions that improve the model precision.

2.2.3 Particle Swarm Optimization

Particle Swarm Optimization (PSO) is a commonly used algorithm [13]. The method is inspired by flocking behavior, where particles or "agents" search through the limitations to find the best suiting values. Each particle adjusts its position based on its own performance and the best performing particle in the swarm. This means that the swarm will converge fast towards the best found values fast and search in that area. However, PSO may converge to local optima in highly nonlinear identification problems.

2.2.4 Fossen modeling

The Fossen model is derived directly from the Newton-Euler equations and follows the structured formulations commonly used in marine control systems. A more detailed description is provided in subsection 2.1.2. The model is well suitable for low-speed motion, autonomous vessel control, and state estimation. Although the parameterization is relatively compact, the inclusion of added mass and nonlinear damping terms can complicate parameter identification. However, accurate calibration requires a well designed system identification procedure.

2.2.5 Abkowitz modeling

The classical Abkowitz model describes the hydrodynamic forces and moments acting on a vessel as a function of surge velocity, sway velocity, yaw rate and rudder angle. Through Taylor-series expansion around a steady operating point, the model approximate the forces and moments in surge X , sway Y , and yaw N .

Although the Abkowitz model provide a detailed description of maneuvering dynamics, it typically involves a large number of hydrodynamic coefficients, often exceeding 60 hydrodynamic coefficients. This detailed parameterization is advantageous when extensive physical insight and high-quality experimental data are available. However, such information is often unavailable when developing generalized identification methodologies. A more detailed formation of the model is provided in subsection 2.1.3.

2.2.6 Modified Abkowitz modeling

To address the complexity of the full Abkowitz model, a modified formulation is used to reduce the number of parameters. The modified model captures the essential dynamics required for low-speed maneuvering.

In this work, a set of 17 hydrodynamic parameters is selected based on:

- Symmetry considerations for conventional monohull geometry.
- Limited maneuvering data available for hydrodynamic observation.
- Simplified to fit general models of leisure vessels.

Although the modified Abkowitz model still requires certain vessel-specific properties, the reduced parameter set simplifies the identification process. In practice, a subset of these parameters, together with actuator and bias terms, is identified using optimization-based methods.

2.3 Path planning

As part of an autonomous docking algorithm, path planning is needed to guide the system to desired states.

2.3.1 A*

A* is a well known and widely used shortest path algorithm originating from computer science. The algorithm was developed with the purpose of optimally finding paths through weighted graphs first introduced in 1968 [14]. Given that a weighted graph includes a source node and a goal node, the algorithm manages to find the shortest path with respect to the assigned weights. A* is a best-first search and an informed search algorithm using heuristic information to guide its search. This forces the algorithm to choose the lowest next node for each step, contributing to its good efficiency compared to an uninformed algorithm such as the Dijkstra search algorithm. The system has a priority queue that tracks the cost to un-explored nodes and its estimate to the goal node.

A* aims to minimize $\hat{f}(n)$ where $f(n)$ is an optimal path through node n . The function $f(n)$ is described as the sum of two parts according to Equation 2.34.

$$f(n) = g(n) + h(n). \quad (2.34)$$

Here, $g(n)$ is the actual cost of an optimal path from the source node to the n th node, and $h(n)$ is the estimated cost of an optimal path from node n to the goal node [14]. This can be used to produce the shortest path in a marine setting, including obstacles to produce viable path points for a controller.

The Python package **extremitypathfinder** utilizes the A* algorithm for Python applications of optimal graph path generation. For this to be applicable each obstacle need to be modeled up as individual polygons.

2.3.2 Interpolation

The optimal path generated from the A* algorithm need to be extended to be applicable for a way-point reference system. This is done by an interpolation algorithm that computes with fixed distance additional points along the optimal vertices.

```

INTERPOLATE PATH(path, s) :  out = [ ]
                             for  $i = 0, \dots, |\text{path}| - 2$  :
                                $p_1 \leftarrow \text{path}[i]$ ,  $p_2 \leftarrow \text{path}[i + 1]$ 
                                $d \leftarrow \|p_2 - p_1\|$ ,  $n \leftarrow \lfloor d/s \rfloor$ 
                               append(out,  $p_1$ )
                               if  $n > 0$  :
                                  $\Delta \leftarrow (p_2 - p_1)/(n + 1)$ 
                                 for  $j = 1, \dots, n$  :
                                   append(out,  $p_1 + j\Delta$ )
                               append(out, path[-1])
                             return out

```

2.3.3 Path smoothing

The interpolated path consists of piecewise-linear segments which introduce heading discontinuities at each vertex. To obtain a C^2 -continuous reference suitable for the MPC, a Cubic Spline [15] is fitted through the interpolated way points. Given the way point sequence $\{(x_i, y_i)\}_{i=0}^n$, a parameter $t_i = i$ is assigned to each point. Two independent natural cubic splines $S_x(t)$ and $S_y(t)$ are constructed such that

$$S_x(t_i) = x_i, \quad S_y(t_i) = y_i, \quad i = 0, \dots, n \quad (2.35)$$

with continuous continuity conditions S_x'' and S_y'' in every interior knot. The smoothed path is then sampled at $m = n \cdot n_{\text{int}}$ uniformly-spaced parameter values:

$$t_j^* = \frac{j(n)}{m-1}, \quad j = 0, \dots, m-1 \quad (2.36)$$

yielding a dense and smooth reference $\{(S_x(t_j^*), S_y(t_j^*))\}$.

2.4 Model Predictive Control

Model Predictive Control (MPC) is described as a control strategy in which the current control action is obtained by solving at each sample step a finite horizon open-loop optimal control problem [16]. The intended application of MPC in this thesis is high-level trajectory tracking and control of a marine vessel during autonomous docking maneuver. Following the results in [2] the control inputs correspond to the achievable motions of the vessels in surge, sway, and yaw.

The controller relies on a mathematical model of the vessel dynamics and utilizes current available state information together with a set of constraints to compute an optimal control action. In addition to determine the immediate control input,

MPC predicts future system behavior over the predefined prediction horizon and generates a sequence of future control actions. Only the first control action of this sequence is applied, after which the optimization problem is solved again at the next sampling instant in a receding horizon fashion.

Rawlings, Mayne, and Diehl describe the principle of MPC as follows: “*The basic concept of MPC is to use a dynamic model to forecast system behavior and optimize the forecast to produce the best decision—the control move at the current time.*” [17]. The optimal control action depends on the initial state of the system and is based on the previous record of measurements.

2.4.1 Dynamic models

A key component in the development of an MPC is the availability of a mathematical model of the control problem. The accuracy of the controller is heavily dependent on the quality of the model. This usually leads to extensive tuning of the model and the associated optimization problem. The model complexity including the dynamics and number of states have a large impact on the computational complexity, where the horizon will accumulate this. Concerning the autonomous docking, two different models will be compared. The Fossen dynamical 3-DOF model subsection 2.1.2 and modified Abkowitz 3-DOF model subsection 2.1.3.

In [2], the MPC formulation is based solely on a kinematic model, which limits its applicability when aiming to design a controller suitable for realistic simulations and real-world deployment. To bridge this gap, system dynamics must be incorporated into the MPC formulation. To achieve this, the 3-DOF modified nonlinear dynamic model by Abkowitz, proposed in [7] is used. Where a simplification of the dynamics of the vessel and hydrodynamic effects of the surrounding environment are taken into account, as described in Equation 2.5.

2.4.2 Discretization

Most MPC applications are modeled after physical systems and are therefore naturally continuous processes.

$$\frac{dx}{dt} = f(x, u). \quad (2.37)$$

This continuous process optimal control problem need to be discretized to suit the format of a digital computer. This is achieved by sampling the continuous dynamics at fixed time intervals and applying a control action based on the available measurements. The commonly used fourth-order Runge-kutta method (RK4) can be used for this application [17], which provides an accurate approximation for ordinary differential equations. RK4 is an explicit integration scheme that evaluates the system dynamics multiple times within each time step, yielding high accuracy while maintaining relatively low computational cost. The method is defined as

$$\begin{aligned}
k_1 &= f(t, x), \\
k_2 &= f(t + h/2, x + (h/2)k_1), \\
k_3 &= f(t + h/2, x + (h/2)k_2), \\
k_4 &= f(t + h, x + hk_3), \\
\Phi &= (h/6)k_1 + (h/3)k_2 + (h/3)k_3 + (h/6)k_4.
\end{aligned} \tag{2.38}$$

For the MPC the RK4 is used to compute the discrete time state update of x_{k+1} by combining the four slope approximations $k_1 - k_4$ over a timestep Δt . Previous work has employed the RK4 scheme on similar problems where it has a valid trade-off between computational complexity and numerical accuracy compared to simpler schemes such as Euler integration [18].

2.4.3 Nonlinear Model Predictive Control

Nonlinear Model Predictive Control (NMPC) extends linear MPC by incorporating nonlinear system models, non-quadratic cost functions and general nonlinear constraints [19]. In continuous-time an NMPC can be formulated as [20]

$$\min_u \int_{t_0}^{t_0+T} \|x(t) - x_r(t)\|_{Q_x}^2 + \|u(t) - u_r(t)\|_{Q_u}^2 dt + \|x(t_0+T) - x_r(t_0+T)\|_{Q_f}^2 \tag{2.39}$$

subjected to:

$$x(t_0) = x_0, \tag{2.40}$$

$$\dot{x}(t) = F(x(t), u(t)), \quad \forall t \in [t_0, t_0 + T] \tag{2.41}$$

$$g(t, x(t), u(t)) \leq 0, \quad \forall t \in [t_0, t_0 + T]. \tag{2.42}$$

Where $Q_x > 0$, $Q_u > 0$, and $Q_f > 0$ are weighting matrices. The state vector is denoted by $x \in \mathbb{R}^{n_x}$, the control input by $u \in \mathbb{R}^{n_u}$, and x_0 represents the current initial state of the system. x_r and u_r are the desired reference trajectories for system state and control input. To enable numerical solution of the continuous-time Equation 2.39, the function needs to be discretized according to subsection 2.4.2. The discrete-time NMPC can be formulated as:

$$\min_{U_i} \sum_{k=0}^{N-1} \|x_{i+k} - x_{i+k}^{ref}\|_{Q_x}^2 + \|u_{i+k} - u_{i+k}^{ref}\|_{Q_u}^2 + \|x_{i+N} - x_{i+N}^{ref}\|_{Q_f}^2 \tag{2.43}$$

subject to:

$$x_i = x_0 \tag{2.44}$$

$$x_{i+k+1} = f_d(x_{i+k}, u_{i+k}) \quad k = 0, \dots, N-1 \tag{2.45}$$

$$g_i(x_{i+k}, u_{i+k}) \leq 0, \quad k = 0, \dots, N-1 \tag{2.46}$$

2.4.4 Cost functions

In MPC, cost functions are formulated as mathematical expressions that the controller seeks to minimize in order to compute an optimal control action. These cost functions encode what the controller interprets as desirable and undesirable behavior by penalizing deviations from desired performance criteria. Therefore, weighting has a significant impact on the resulting control behavior.

2.4.4.1 Cross-track error

Keeping the vessel on the reference track is essential for an autonomous vessel. The reference track is generated by a shortest path algorithm section 2.3. To follow the intended track, a cross-track error (CTE) is formulated and used as a cost function within the MPC framework. This cost function penalizes deviations between the marine vessel's center of gravity and the reference path. Two consecutive points on the reference path, s_1 and s_2 , as illustrated in Figure 2.3 are used. The vessel position $\mathbf{p}$ can be projected onto the line segment connecting these points to find the closest point on the segment. This is achieved by computing a projection parameter $\hat{t}$ as:

$$\hat{t} = \frac{(\mathbf{p} - s_1) \cdot (s_2 - s_1)}{\|s_2 - s_1\|^2}. \quad (2.47)$$

To ensure that the projection lies within the segment, $\hat{t}$ is clamped to the interval $[0,1]$

$$t^* = \max(0, \min(1, \hat{t})) \quad (2.48)$$

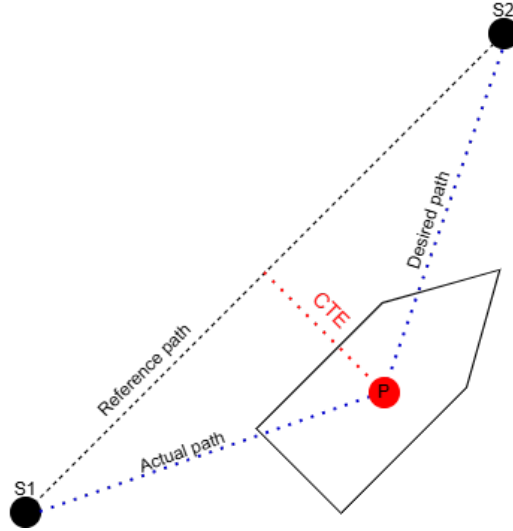


Figure 2.3: CTE illustration.

The distance between the vessel and the line segment is calculated as Equation 2.49.

$$\mathbf{D} = s_1 + t^*(s_2 - s_1) - \mathbf{p}. \quad (2.49)$$

The CTE cost function is formulated in Equation 2.50.

$$\mathbf{J}_{CTE}(\mathbf{p}) = Q_{CTE}(\min(\|\mathbf{p} - s_1 - t^*(s_2 - s_1)\|^2)) \quad (2.50)$$

where Q_{CTE} is a tuning weight that determines the relative importance of minimizing the cross-track deviation.

2.4.4.2 Control input/ acceleration

To reduce jerk in the system a cost was introduced to penalize quick changes in control inputs. The control input is set as $u=[F_x, F_y, N_z]$ with the following cost function:

$$J_{\text{control}} = Q_{\text{acc}} \|\mathbf{u}_{k+1} - \mathbf{u}_k\|^2, \quad k \in \mathcal{N}_{0, N_u-1}. \quad (2.51)$$

In Equation 2.51 u represents the control input vector. Q_{acc} represents penalty weights subject to the control inputs X, Y and N. The cost function penalizes changes control inputs to give the MPC smooth transitions and reduce quick changes in the system.

2.4.4.3 Terminal cost

The goal of the autonomous docking procedure is to arrive at a set goal state x_{goal} from an initial state x_{init} . The state is described in Equation 2.52.

$$\begin{aligned} x_{\text{init}} &= \begin{bmatrix} x \\ y \\ \psi \\ u \\ v \\ r \end{bmatrix}, \\ x_{\text{goal}} &= \begin{bmatrix} x_g \\ y_g \\ \psi_g \\ u_g \\ v_g \\ r_g \end{bmatrix}. \end{aligned} \quad (2.52)$$

Docking at the right position and orientation is crucial for a successful maneuver. The velocity states u, v and r are naturally set to zero as the terminal state. The cost function is formulated as Equation 2.53

$$\mathbf{J}_{\text{terminal}} = Q_{\text{pos}}((\mathbf{x} - \mathbf{x}_r)^2 + (\mathbf{y} - \mathbf{y}_r)^2) + Q_{\psi}(\psi - \psi_r)^2 \quad (2.53)$$

Q_{pos} and Q_{ψ} are weights applied to the position and angle error. The terminal state cost is not expected to be active during the initial path follower, rather it will be activated when the *docking_mode* flag is activated close to the final position. Excluding this cost during the path following allows the system to better align with the nearby references in position and angle without being influenced by the terminal state.

2.4.5 Velocity cost

A velocity cost is introduced to regulate vessel motion. The sway velocity is penalized to reduce lateral motion, while the surge velocity is constrained to avoid excessive forward speed.

The cost function is defined as:

$$J_{\text{vel}} = Q_u \cdot \max(0, (u - u_{\text{desired}})^2) + Q_v \cdot v^2. \quad (2.54)$$

The surge term is asymmetric and only penalizes velocities exceeding the desired cruising speed. This allows the vessel to operate below the desired speed without penalty, while preventing excessive forward motion.

2.4.6 Heading cost

To keep the system on the right heading track a reference was computed based on the pre-determined path. This was then utilized to penalize deviations from the reference heading. The cost function is formulated in Equation 2.56.

$$e_\psi = \text{atan2}(\sin(\psi - \psi_{\text{ref}}), \cos(\psi - \psi_{\text{ref}})) \quad (2.55)$$

$$J_{\text{heading}} = Q_{\text{Heading}} \max(0, e_\psi^2 - \theta_{\text{th}}^2) \quad (2.56)$$

The $\text{atan2}(\sin(\cdot), \cos(\cdot))$ formulation wraps the angular error to the interval $(-\pi, \pi)$, thereby avoiding discontinuities at $\pm\pi$. The parameter θ_{th} defines a dead-band for yaw deviations.

2.4.6.1 Obstacle and boundary avoidance

To ensure safe navigation, several cost terms are introduced to penalize unsafe states. These include static obstacle avoidance, boundary constraints, and dynamic obstacle avoidance.

The static obstacle cost penalizes configurations where the vessel enters forbidden regions, defined using polygonal representations of obstacles. This is formulated as

$$J_{\text{obs}} = Q_{\text{obs}} \cdot f_{\text{inside}}(x, y), \quad (2.57)$$

where $f_{\text{inside}}(\cdot)$ is a function that evaluates whether the vessel is inside a polygonal obstacle.

Similarly, boundary constraints are enforced using a penalty function

$$J_{\text{boundary}} = Q_{\text{bound}} \cdot f_{\text{boundary}}(x, y), \quad (2.58)$$

which penalizes violations of the operational area.

To account for moving obstacles, a dynamic obstacle cost is introduced,

$$J_{\text{dyn}} = Q_{\text{dyn}} \cdot f_{\text{dynamic}}(x, y, t), \quad (2.59)$$

where the obstacle position is predicted over the MPC horizon.

These terms ensure collision avoidance and safe operation within constrained environments.

2.4.7 Optimization engine

The optimization engine opEN was used to solve the optimal control problem. The solver is open source and is a Rust-based solver ideal for embedded applications. OpEN combines fast numerical optimization methods with the fast and safe programming language Rust. The solver is said to be efficient for highly dynamic non linear systems and MPC calculations [21]. The problem that is solved is formulated in Equation 2.60.

$$\begin{aligned}
\mathbf{P}(p) : & \underset{u \in R^{nu}}{\text{Minimize}} f(u, p) \\
& \text{subject to } u \in U \\
& F_1(u, p) \in C \\
& F_2(u, p) = 0
\end{aligned} \tag{2.60}$$

The minimization problem is then solved by the proximal averaged Newton-type method (PANOC) [21]. The PANOC was first presented in 2017 in [22] where it was introduced as an efficient method to solve nonlinear constrained optimal control problems. The combination of line-search forward-backward iterations with fast Newton-type steps results in a simple, fast and convergent algorithm. According to [21] this is seen as a good solver for embedded applications.

2.5 Disturbance handling

This section contains theory regarding the disturbance modeling and handling needed for a robust control of a marine vessel. This is a crucial step towards autonomy concerning the harsh reality of marine environment and measurement uncertainties.

2.5.1 Kalman Filter

The Kalman Filter (KF) can be used to predict future states based on previous measurements and states. The filter estimates the current state using the measured data and the prediction step to obtain an accurate model representation. The KF is a model-based method that can be used to filter measurements and in a sensor fusion manor predict better estimates. Another key component is the ability to estimate and predict hidden states such as disturbances.

2.5.2 Extended Kalman Filter

Since the KF only works on linear systems, an Extended Kalman Filter (EKF) is used to capture the nonlinearities that are present in the marine model. After the initial estimation of the states, the discrete time EKF solves the following system:

$$x_k = f(x_{k-1}, u_{k-1}) + w_{k-1}, \tag{2.61}$$

$$z_k = h(x_k) + v_k, \tag{2.62}$$

where $x = [x, y, \psi, u, v, r]^T$ is the state vector. The function f is a nonlinear function of the state vector x_k . The term z_k is the measurement information using the available data, where h_k is a vector function mapping the observations. The noise terms w_k and v_k are process and observation noise, independent of each other and

Gaussian white noise with zero mean and covariance matrices Q and R . The EKF begins with linearization and computation of the Jacobians.

$$F_k = \left. \frac{\partial f}{\partial x} \right|_{\hat{x}_{k-1|k-1}, u_{k-1}}, \quad H_k = \left. \frac{\partial h}{\partial x} \right|_{\hat{x}_{k|k-1}} \quad (2.63)$$

Prediction step:

$$\hat{x}_{k|k-1} = f(\hat{x}_{k-1|k-1}, u_{k-1}) \quad (2.64)$$

$$P_{k|k-1} = F_k P_{k-1|k-1} F_k^T + Q_{k-1} \quad (2.65)$$

Update step:

$$\tilde{y}_k = z_k - h(\hat{x}_{k|k-1}) \quad (2.66)$$

$$K = P_{k|k-1}^- H_k^T (H_k P_{k|k-1}^- H_k^T + R_k)^{-1} \quad (2.67)$$

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k \tilde{y}_k \quad (2.68)$$

$$P_{k|k} = (I - K_k H_k) P_{k|k-1} \quad (2.69)$$

The state vector can be described as:

$$x_k = [x, y, \psi, u, v, r]^T. \quad (2.70)$$

For online disturbance estimation, the augmented state vector includes slowly-varying NED-frame disturbance force b_N and b_E and disturbance moment b_r . The biases are modeled as random walks with no deterministic dynamics, which is what makes the EKF able to track slowly varying environmental forces like wind and current.

$$\hat{x}_k = [x, y, \psi, u, v, r, b_N, b_E, b_r]^T. \quad (2.71)$$

Since the bias states are initially unknown, the EKF starts with:

$$\hat{x}_0 = [x_0, y_0, \psi_0, u_0, v_0, r_0, 0, 0, 0]^T. \quad (2.72)$$

The states update through the filter due to the Kalman gain:

$$\hat{x}_{k,\text{new}} = \hat{x}_{k,\text{old}} + K_k (z_k - H \hat{x}_{k,\text{pred}}). \quad (2.73)$$

An overview of the interaction between the MPC and EKF is shown in Figure 2.4.

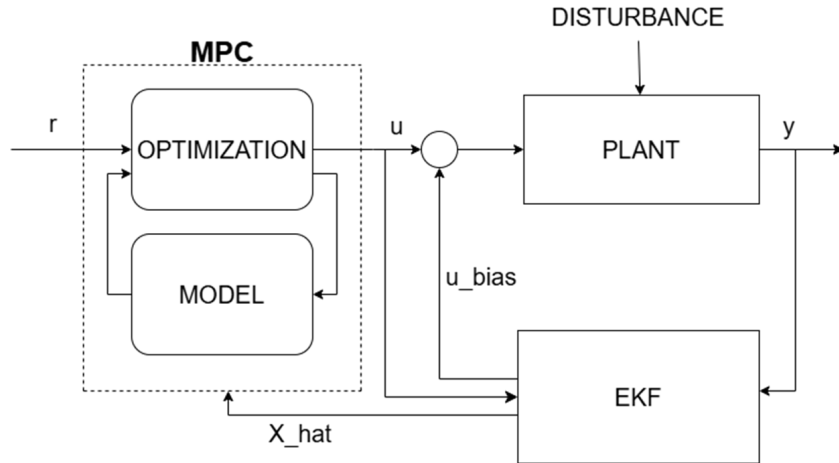


Figure 2.4: EKF inside a control loop.

One major challenge with KF is model dependency. The outcome is based on the internal mathematical model and the quality of the input data. A KF requires more computational effort than simple filtering methods. The input data must be accurately calibrated and the noise distributions must be known or estimated for good estimations.

2.6 Control allocator

Control allocation is a low-level control principle that is useful for control of over-actuated systems. The allocator deals with distributing the total control demand among the available actuators, which makes the actuator selection task separated from the regulation task in the control design [23]. It can also minimize the usage of each actuator leading to less wear and tear and optimizing efficiency.

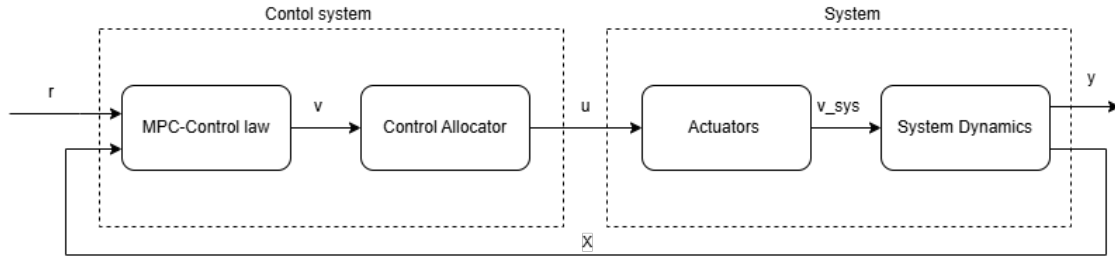


Figure 2.5: Control system structure when control allocation is used.

A system utilizing a control allocator is illustrated in Figure 2.5. For a marine vessel, the different actuators are the methods of propulsion of the vessel, including the direction of the rudders. The implementation of an allocator is suitable because the high-level MPC does not deal with the specific characteristics of each available actuator. If the system is over-actuated, there will be multiple combinations of actuator inputs that will yield the same global solution, and therefore a way to choose the optimal is needed. Another possible design feature that uses an allocator is the fault-tolerant control of multiple actuators, which increases the redundancy and robustness of the system [24].

An actuator system can be written as :

$$\boldsymbol{\tau} = \mathbf{B}u \quad (2.74)$$

where $\boldsymbol{\tau}$ is the desired force and moment (surge, sway, and yaw). u is the actuator commands, the motor thrusts, and $\mathbf{B}$ is the control effectiveness matrix.

For marine vessels in n DOF it is necessary to distribute the generalized control forces $\boldsymbol{\tau}$ to the actuators in terms of control input angles α and rpm of each thruster to produce the desired force.

3

Methods

This chapter presents the methods used to design, implement and evaluate the autonomous docking system developed in this project. The workflow begins with model-based control development using a reduced-order nonlinear maneuvering model. The model is identified using experimental and simulated vessel data, with the model parameters estimated through a combination of structured maneuvers and optimization-based system identification. The identified dynamic model is then integrated into a NMPC framework for trajectory tracking, obstacle avoidance, and docking. The NMPC computes the desired generalized forces and moments in surge, sway, and yaw. To compensate for model mismatches and external disturbances, an EKF is employed for state estimation and disturbance compensation. Finally, a control allocator, decoupled from the high-level controller, maps the generalized force commands to actuator inputs while respecting vessel and actuator constraints. The complete control architecture is validated through simulations and experiments on a physical vessel.

3.1 System identification

System identification is applied to characterize how the vessel's dynamics respond to control inputs and environmental disturbances. The primary objective is to identify the hydrodynamic characteristics for low-speed maneuvering and to capture the interaction between the marine vessel and its surrounding environment. By performing predefined maneuvers, the vessel's dynamic response to thrust and steering inputs can be observed and analyzed.

3.1.1 Step-response gray-box

A pragmatic step-response-based identification approach is used to obtain control-relevant vessel parameters. The primary aim of this method is to capture the dominant low-speed dynamics while maintaining a model structure suitable for control applications such as MPC. The initial calibration step in identifying the unknown dynamics of the monohull vessel is to perform controlled and predictable step-response logging. These experiments are later conducted either on a physical vessel or in a simulated (dummy) environment.

The step-response applies a constant, known control input until a steady state is reached, and these step-responses are applied to the surge, sway, and yaw degrees of freedom. This procedure contributes to a gray box identification approach, where the different degrees of freedom are decoupled. For this system, a predefined rudder angle and engine speed command are sent over the CAN-network. In steady state,

the control inputs are set to zero to achieve a coasting down behavior. An overview of the calibration step-response can be seen in Figure 3.1. During logging, the following states are recorded over time:

1. GPS position
2. Speed over ground
3. Heading
4. Engine RPMs
5. Rudder angles

The steady state of the step-response is then used to derive the terminal velocities corresponding to a constant actuator input. In steady state, thrust is assumed to balance hydrodynamic drag, and the assumption of the nonlinear propeller curve is given by Equation 3.1.

$$[T_i] = [k_1 \cdot rpm_i + k_2 \cdot |\frac{rpm_i}{1000}|^2] \quad (3.1)$$

The maximum achievable forces in the body frame of the marine vessel can then be derived in Equation 3.2, where T_i is the maximum thrust from Equation 3.1, n is the number of actuators, δ is the maximum actuator angle, and $L_{y,i}$ is the length of the lever arm from the center of rotation.

$$\begin{aligned} X_{max} &= \sum_{i=1}^n T_{surge} \\ Y_{max} &= \sum_{i=1}^n T_{sway} \sin(\delta_{max}) \\ N_{max} &= \sum_{i=1}^n L_{y,i} T_{yaw} \end{aligned} \quad (3.2)$$

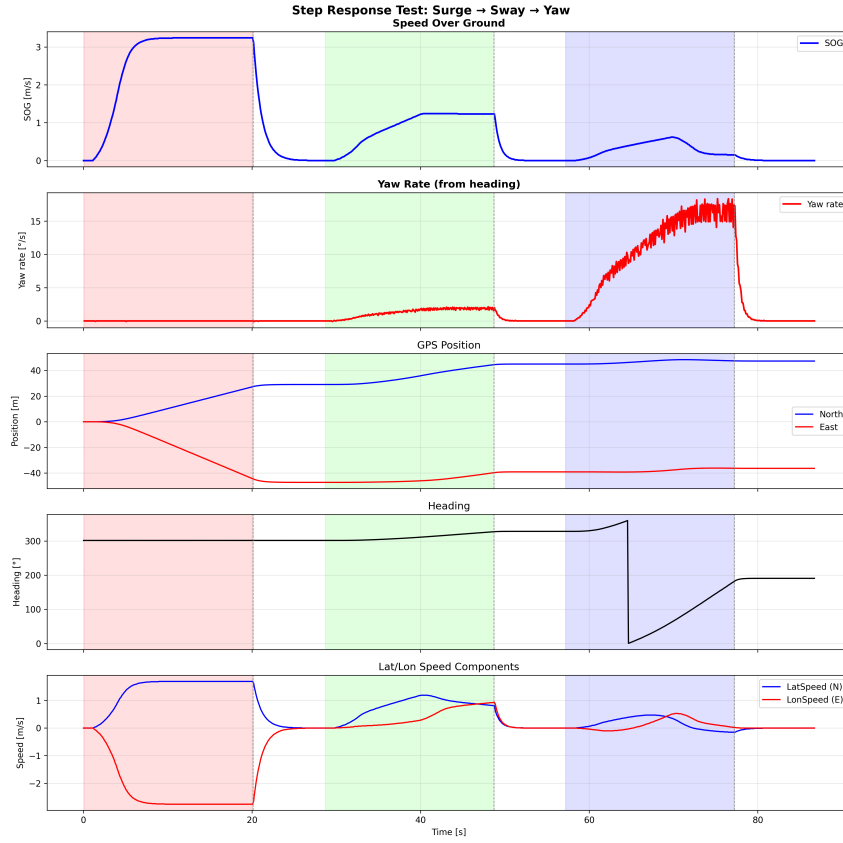


Figure 3.1: Calibration Step-response, on the dummy simulation station.

The final gray-box estimate is performed in two steps to estimate the effective mass, which should not be confused with the actual mass of the vessel. The effective mass represents combined rigid-body and added-mass effects and is decoupled in each degree of freedom using the step-response calibration data. The estimation is carried out using two separate methods. First, the initial acceleration during the ramp-up phase is estimated using a linear least-squares fit.

$$m_i = \frac{F}{a_{initial}} \quad (3.3)$$

$$\begin{aligned} m_{11}^{(accel)} &= \frac{X_{\max}}{a_{\text{surge}}} \\ m_{22}^{(accel)} &= \frac{Y_{\max}}{a_{\text{sway}}} \\ m_{33}^{(accel)} &= \frac{N_{\max}}{\dot{r}_{\text{initial}}} \end{aligned} \quad (3.4)$$

The second method for estimating the effective mass uses the coast-down decelerations. After the constant-thrust phase, the control input is set to zero and the vessel decelerates under hydrodynamic drag alone. At this stage, the thrust and drag are seen as equal since steady state. A linear least square over the first seconds of the coast phase produces the initial deceleration d_0 , and the same force-balance as in Equation 3.4 yields Equation 3.6

$$m_{\text{eff}}^{(\text{coast})} = \frac{F_{\text{drag}}(v_{\text{max}})}{|d_0|} \quad (3.5)$$

$$\begin{aligned} m_{11}^{(\text{coast})} &= \frac{X_{\text{max}}}{d_{\text{surge}}} \\ m_{22}^{(\text{coast})} &= \frac{Y_{\text{max}}}{d_{\text{sway}}} \\ m_{33}^{(\text{coast})} &= \frac{N_{\text{max}}}{d_{\text{yaw}}} \end{aligned} \quad (3.6)$$

These two methods are based on the same thrust but then independent measurement giving value to the effective mass estimation. The final effective mass estimate is the sum of the two weighted estimates in Equation 3.7.

$$m_i = w_{\text{accel}} m_i^{(\text{accel})} + w_{\text{coast}} m_i^{(\text{coast})} \quad (3.7)$$

The estimated parameters are used as the foundation for the other hydrodynamic coefficients.

3.1.2 Zigzag maneuver

The second calibration stage is a zigzag maneuver designed to find the coupled surge and yaw dynamics during turning. The angle of the rudder and the engine speed are sent over the CAN-network to produce a predefined zigzag behavior as illustrated in Figure 3.2. The heading alternates between ± 20 deg and the surge speed is set to constant. The data set is logged and used for identification of the hydrodynamic parameters.

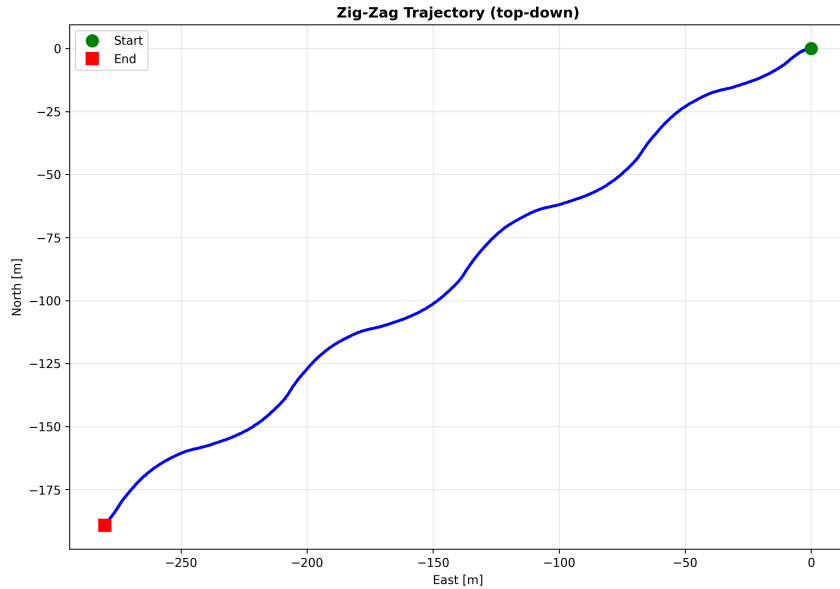


Figure 3.2: 20/20 zigzag trajectory.

3.1.3 Drag constant identification surge

Isolating the surge DOF is required to identify the drag constants independently, which minimizes cross-coupling effects during hydrodynamic identification. To achieve

this, acceleration and speed measurements straight-line in surge motion are required. When isolating this, the formula in Abkowitz becomes the following:

$$m\dot{u} = \underbrace{mvr}_{v \approx 0} + \underbrace{mx_g r^2}_{r \approx 0} + X_H + X_{ctrl} + X_0, \quad (3.8)$$

$$X_H = X_u u + X_{u|u}|u| \rightarrow \underbrace{-r_1}_{X_u} + \underbrace{(-r_2)u|u|}_{X_{u|u}} + X_0, \quad (3.9)$$

where r_1 is the linear damping, r_2 is the quadratic damping, and X_0 is the constant bias of the conditions in the water when performing the tests. This can then be written as the following drag force formula:

$$F_k = -m\dot{u}_k \rightarrow \begin{bmatrix} F_1 \\ F_2 \\ \vdots \\ F_N \end{bmatrix} = \begin{bmatrix} u_1 & u_1^2 & 1 \\ u_2 & u_2^2 & 1 \\ \vdots & \vdots & \vdots \\ u_N & u_N^2 & 1 \end{bmatrix} \begin{bmatrix} r_1 \\ r_2 \\ X_0 \end{bmatrix}, \quad (3.10)$$

rewrite this to mass and acceleration terms:

$$\begin{bmatrix} -ma_1 \\ -ma_2 \\ \vdots \\ -ma_N \end{bmatrix} = \begin{bmatrix} u_1 & u_1^2 & 1 \\ u_2 & u_2^2 & 1 \\ \vdots & \vdots & \vdots \\ u_N & u_N^2 & 1 \end{bmatrix} \begin{bmatrix} r_1 \\ r_2 \\ X_0 \end{bmatrix}. \quad (3.11)$$

With this formula and the way to identify the surge constants r_1 and r_2 , some assumptions are needed. The Mass of the vessel needs to be known or set to a specific value. However, estimating the weight of the vessel is challenging, and access to weighting equipment may be limited. To compensate for this estimate, many different tests with different weights of the vessel are performed.

With this method, the span of the mass is tested and validated. Using non-negative least squares, it is possible to minimize the error in the equation and find suitable values on the damping constants and bias term.

3.1.4 Hyperparameter identification

With the surge drag parameter X_u determined, the remaining model parameters must now be identified. This is approached in a consistent manner that preserves relationships among the parameters. The parameters that need to be identified are the following:

$$Sway = Y_v, Y_r, Y_d, \quad (3.12)$$

$$Yaw = N_r, N_{rr}, N_v, N_d, \quad (3.13)$$

$$propeller = k_1, k_2, ly_t. \quad (3.14)$$

To limit the search for GA, bounds are set for the geometry of the hull and the surge parameters X_u . In addition, physics-based scaling relationships are introduced to

constrain the remaining hydrodynamic coefficients. These relationships are based on empirical observations and standard maneuvering theory, and ensure that the identified parameters remain within physically realistic ranges:

$$Y_v \approx (2.8) * |X_u| \quad (3.15)$$

$$N_r \approx |Y_v| * \left(\frac{L}{2}\right)^2 \quad (3.16)$$

$$N_v \approx |Y_v| * \left(\frac{L}{4}\right) \quad (3.17)$$

$$Y_r \approx |Y_v| * \left(\frac{L}{4}\right) \quad (3.18)$$

$$N_{rr} \approx (0.3) * |N_r| \quad (3.19)$$

$$ly_t \in \left(0.3, \frac{B}{2}\right) \quad (3.20)$$

$$|X_{uu}| < 3 * \frac{|X_u|}{u_{max}} \quad (3.21)$$

These constraints reduce the dimensionality of the feasible parameter space and guide the GA toward physically meaningful solutions. This improves convergence behavior and helps prevent the identification of non-realistic parameter sets while still allowing sufficient flexibility to accurately reproduce the vessel dynamics.

3.1.5 Implementing the vessel model

Based on an evaluation of available maneuvering models, the Modified Abkowitz Model (MAM) was selected as the most suitable representation for the dynamics of the vessel considered in this thesis. The model provides an appropriate balance between physical fidelity and computational efficiency, making it well suited for low-speed maneuvering and docking operations. The MAM equations are represented in subsection 2.1.4.1. The structure of MAM allows for parameterization of hydrodynamic forces and moments, enabling tuning based on the measured or simulated responses of the marine vessel. Together with the GA, it is possible to extract parameter information in the model that suits the dynamics of a vessel following specific inputs and outputs.

The vessel is actuated by two control units: port thruster (P) and starboard thruster (S). The thrust generated by each thruster is modeled as a nonlinear function of the propeller shaft speed:

$$\begin{bmatrix} T_P \\ T_S \end{bmatrix} = \begin{bmatrix} k_1 \cdot rpm_P + k_2 \cdot \left|\frac{rpm_P}{1000}\right|^2 \\ k_1 \cdot rpm_S + k_2 \cdot \left|\frac{rpm_S}{1000}\right|^2 \end{bmatrix}. \quad (3.22)$$

The thruster forces are then transformed into global surge, sway, and yaw moments using the actuator configuration matrix B :

$$\begin{bmatrix} T_X \\ T_Y \\ T_N \end{bmatrix} = \underbrace{\begin{bmatrix} \cos(\delta_P) & \cos(\delta_S) \\ \sin(\delta_P) & \sin(\delta_S) \\ l_x \sin(\delta_P) + l_y \cos(\delta_P) & l_x \sin(\delta_S) - l_y \cos(\delta_S) \end{bmatrix}}_B \begin{bmatrix} T_P \\ T_S \end{bmatrix}, \quad (3.23)$$

where l_x and l_y denote the longitudinal and lateral distances from the vessel's center of gravity (CoG) to each thruster. These forces and moments are added to the hydrodynamic contribution in the MAM. The total forces and moment acting on the vessel are expressed as:

$$X = X_u u_s + X_{uu} |u_s| u_s + T_X, \quad (3.24)$$

$$Y = Y_v v_s + Y_r r_s + Y_\delta \delta + T_Y, \quad (3.25)$$

$$N = N_r r_s + N_{rr} |r_s| r_s + N_v v_s + N_\delta \delta + T_N, \quad (3.26)$$

where u_s , v_s , and r_s denote body fixed surge velocity, sway velocity, and yaw rate. The vessel accelerations are computed using the coupled equations of motion:

$$\dot{u} = \frac{X + (m_{22} v_s + m_{23} r_s) r_s}{m_{11}}, \quad (3.27)$$

$$y_{\text{rhs}} = Y - m_{11} u_s r_s, \quad (3.28)$$

$$n_{\text{rhs}} = N - m_{xg} u_s r_s, \quad (3.29)$$

$$\dot{v} = \frac{m_{33} y_{\text{rhs}} - m_{23} n_{\text{rhs}}}{\det(M)}, \quad (3.30)$$

$$\dot{r} = \frac{-m_{23} y_{\text{rhs}} + m_{22} n_{\text{rhs}}}{\det(M)}. \quad (3.31)$$

3.1.6 Genetic Algorithm-based parameter estimation

To estimate the remaining hydrodynamic parameters, a population-based optimization approach using GA is adopted. The method is well suited for nonlinear, non-convex parameter estimation where information is limited due to its efficient exploration of the search space. The primary goal of optimization is to minimize the measured vessel motion with the predicted dynamic model. With logged data of propeller rotational speed (RPM), rudder angle, and vessel velocities as input to the model. The output velocities of the model are then directly compared with the measured data. The optimization problem is formulated as:

$$\min_{\theta} \sum_{k=1}^N ||\mathbf{v}_{\text{meas}}(k) - \mathbf{v}_{\text{model}}(k, \theta)||^2, \quad (3.32)$$

where θ denotes the set of hydrodynamic and propulsion parameters to be identified. The individual in the GA population represents a set of candidate parameters. For

each individual, the dynamic vessel model is simulated using the set of parameters with the recorded control inputs, RPM, and rudder angle. The resulting velocities, headings and positioning are then compared with the true measured vessel velocities, headings and positioning. The resulting error defines the fitness score used to select a population reproduction.

3.2 Model Predictive Control

In the paper [2], MPC is formulated using a purely kinematic model. This approach simplifies the controller design and reduces computational complexity. However, it neglects the importance of the dynamic effects and external disturbances, which are necessary to accurately represent the motion of a real marine vessel. As a result, the applicability of such a simplified controller is limited by real-world docking scenarios. To make it as realistic as possible, the implementation of vessel dynamics and environmental forces in the MPC is needed to represent a good framework of realistic operation.

3.2.1 Prediction model

A key component of MPC is the ability to predict the future behavior of the control system over a finite prediction horizon. In this work, the prediction model is based on the modified Abkowitz maneuvering model. The Abkowitz model is commonly used when describing the nonlinear behavior of a marine vessel for low-speed maneuvering applications, such as docking, where nonlinear damping effects have a large impact on the behavior of the vessels. The vessel state vector can be seen in Equation 2.52.

3.2.2 Cost function

The MPC cost function guides the vessel safely and smoothly to the desired docking pose while respecting physical limitations and safety constraints. It penalizes deviations from the reference path, excessive control efforts, abrupt actuator changes, and unsafe maneuvering toward obstacles and boundaries.

$$J = J_{\text{heading}} + J_{\text{CTE}} + J_{\text{vel}} + J_{\text{jerk}} + J_{\text{obs}} + J_{\text{boundary}} + J_{\text{dyn}} + J_{\text{terminal}}. \quad (3.33)$$

The cost function is evaluated over the prediction horizon and minimized at each time step.

During navigation mode, the trajectory tracking terms (J_{CTE} , J_{heading} , and J_{vel}) dominate the behavior of the controller. In contrast, during docking mode, the terminal cost J_{terminal} is activated to ensure convergence to the desired final pose. The obstacle-related cost terms (J_{obs} , J_{boundary} , and J_{dyn}) are active throughout the maneuver to ensure collision avoidance and safe operation.

3.2.3 Solver

The resulting MPC problem is formulated as a constrained nonlinear optimization problem since it involves vessel dynamics, nonlinear damping, and obstacle avoidance. For each control update, the optimization problem is solved in a receding horizon manner.

A nonlinear programming (NLP) solver is used to compute the optimal control sequence. The first control input is only applied to the system, and the sequence is

repeated at the next time step using the updated state measurements. Using this feedback allows the controller to compensate for model mismatches and external disturbances.

To ensure real-time feasibility, the prediction horizon and the solver tolerances are selected based on the available computational time per sampling step.

3.3 Control allocation

The control allocation problem is formulated as a nonlinear mapping from actuator commands to generalized forces acting on the marine vessel. It computes the surge force τ_X , sway force τ_Y , and yaw moment τ_N as a function of the thruster angles and rotational speeds.

The input vector is defined as:

$$u = [\delta_P \quad \delta_S \quad \text{RPM}_P \quad \text{RPM}_S]^T, \quad (3.34)$$

where δ_P and δ_S denote the angles of the port and starboard thrusters, respectively, and RPM_P , RPM_S correspond to the rotational speeds. Positive RPM values represent forward thrust while negative RPM corresponds to reverse thrust.

The thrust generated by each thruster can be described by :

$$T_i = \text{sign}(\text{RPM}_i) \left(k_1 |\text{RPM}_i| + k_2 \left(\frac{|\text{RPM}_i|}{1000} \right)^2 \right), \quad (3.35)$$

where k_1 and k_2 are coefficients identified using the GA. This equation represents the nonlinear propeller characteristics and associated efficiency effects.

The generalized force vector in 3-DOF is given by:

$$\boldsymbol{\tau_d} = \begin{bmatrix} \tau_X \\ \tau_Y \\ \tau_N \end{bmatrix}. \quad (3.36)$$

The surge and sway forces are obtained by describing them in body-fixed frame forces:

$$\tau_X = T_P \cos(\delta_P) + T_S \cos(\delta_S), \quad (3.37)$$

$$\tau_Y = T_P \sin(\delta_P) + T_S \sin(\delta_S). \quad (3.38)$$

The yaw moment is generated by the moment arms that extend from vessel's CoG to the thruster location Table 3.1.

Table 3.1: Thruster positions in the body-fixed coordinate frame.

Thruster	Position	Meaning
Port	$(-l_x, +l_y)$	Behind and to the left (port side) of CoG
Starboard	$(-l_x, -l_y)$	Behind and to the right (starboard side) of CoG

and can be expressed as:

$$\tau_N = l_y(T_P \cos(\delta_P) - T_S \cos(\delta_S)) - l_x(T_P \sin(\delta_P) + T_S \sin(\delta_S)), \quad (3.39)$$

where l_x and l_y denote the longitudinal and lateral distance from the vessel's CoG to the thrusters, respectively, an illustration can be seen in Figure 3.3. Positive yaw moments correspond to clockwise rotation about the CoG in the body-fixed frame. The nonlinear control allocation can be expressed as a mapping from actuator inputs to generalized forces:

$$\boldsymbol{\tau} = h(\mathbf{u}), \quad (3.40)$$

where $h(\mathbf{u})$ represents the nonlinear mapping arising from the trigonometric dependence on thruster angles and the nonlinear thrust-RPM relationship.

The mapping can also be written in terms of thrust forces as:

$$\boldsymbol{\tau} = \mathbf{B}(\boldsymbol{\delta}) \begin{bmatrix} T_P \\ T_S \end{bmatrix}. \quad (3.41)$$

Due to the trigonometric dependence on the thrusters and the nonlinear thruster-RPM relationship, this results in a nonlinear control allocation problem. Given a desired generalized force, see Equation 3.36, the corresponding actuator commands are obtained by solving a nonlinear optimization problem, which is described in [25].

$$\boldsymbol{\tau} = \begin{bmatrix} \tau_X \\ \tau_Y \\ \tau_N \end{bmatrix} = \underbrace{\begin{bmatrix} \cos(\delta_P) & \cos(\delta_S) \\ \sin(\delta_P) & \sin(\delta_S) \\ l_y \cos(\delta_P) - l_x \sin(\delta_P) & -l_y \cos(\delta_S) - l_x \sin(\delta_S) \end{bmatrix}}_{\mathbf{B}(\boldsymbol{\delta})} \begin{bmatrix} T_P \\ T_S \end{bmatrix} \quad (3.42)$$

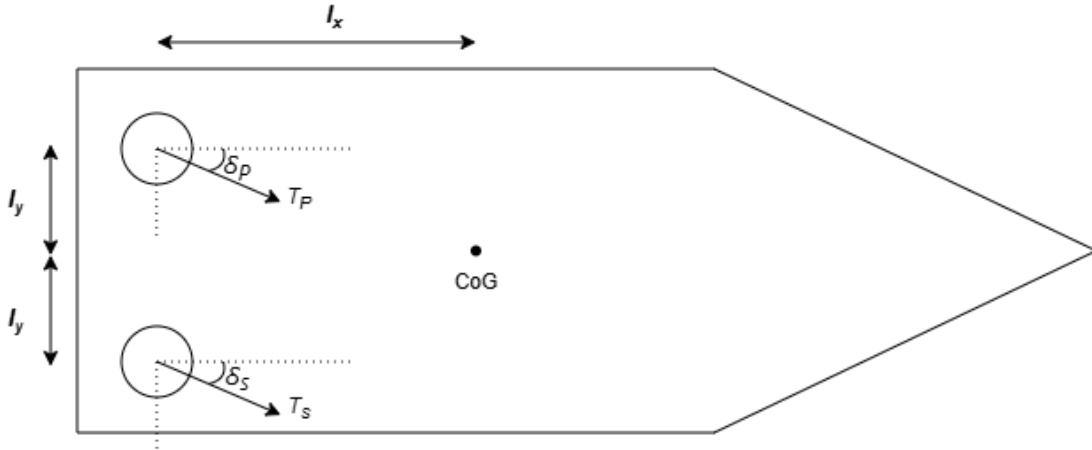


Figure 3.3: Two thrusters, T_P as the thrust from port thruster and T_S as the thrust from starboard thruster. Their location on the marine vessel given with the lengths l_y and l_x .

The allocator is formulated as the following optimization problem:

$$\min_{\mathbf{u}} \left\| \frac{h(\mathbf{u}) - \boldsymbol{\tau}_d}{\boldsymbol{\tau}_{\text{scale}}} \right\|_2^2 + \lambda_P \sum_i \text{RPM}_i^2 + \lambda_R \sum_i \max(-\text{RPM}_i, 0)^2, \quad (3.43)$$

where the first term penalizes the tracking error in force and moment, the second term penalizes the total actuator power, and the third term penalizes reverse thrust.

This penalty is increased in scenarios where forward surge is requested in order to avoid counteracting thrust commands. Additional penalties are introduced to discourage rapid gear switching and changes in RPM direction between consecutive time steps.

$$|\delta_i| \leq \delta_{\max}, \quad (3.44)$$

$$|\text{RPM}_i| \leq \text{RPM}_{\max}, \quad (3.45)$$

$$|\delta_i - \delta_i^{\text{prev}}| \leq \dot{\delta}_{\max} T_s, \quad (3.46)$$

where $\dot{\delta}_{\max}$ denotes the maximum allowed thruster slew rate and T_s is the MPC sampling time. This constraint is motivated by the coupled influence of forces in *surge* and *sway* together with the moment in *yaw*, as discussed in [25]. An additional constraint is imposed on the propeller speeds through an RPM slew-rate limitation:

$$|\text{RPM}_i - \text{RPM}_i^{\text{prev}}| \leq \dot{\text{RPM}}_{\max} T_s, \quad (3.47)$$

which is enforced when the propeller operates in the active thrust region, i.e., $|\text{RPM}_i| \geq \text{RPM}_{\min}$.

The resulting nonlinear optimization problem is solved by using Sequential Least Squares Programming (SLSQP). The optimizer determines the optimal combination to minimize the optimization problem while satisfying the constraints of the system. In non-docking operation, an additional constraint is imposed such that the thruster angles are almost equal, i.e.,

$$\delta_P \approx \delta_S, \quad (3.48)$$

resulting in a symmetric configuration where the yaw motion is primarily achieved through differential thrust. This constraint reflects typical operational conditions and improves efficiency by avoiding conflicting thrust direction.

In docking mode, this constraint is relaxed, allowing the thrusters to work independently to achieve full maneuvering in all degrees of freedom.

3.3.1 General control allocation formulation

This section presents a general control allocation formulation for marine vessels with multiple thrusters. The formulation provides a conceptual framework, while the specific implementation used in this thesis is described in the previous subsection. Since the control allocator is intended for multiple marine vessels, the control effectiveness matrix $\mathbf{B}$ will have different layouts of stern thrusters. The number of thrusters may vary from one to four at the stern.

The generalized forces generated by the thrusters can be described in the following way. Let index i represent the position of the thruster in the body-fixed frame given by

$$\mathbf{r}_i = \begin{bmatrix} l_{xi} \\ l_{yi} \end{bmatrix}. \quad (3.49)$$

The thrust magnitude T_i acts in the direction defined by the steering angle δ_i . The resulting force components in surge and sway are given by:

$$X_i = T_i \cos(\delta_i), \quad (3.50)$$

$$Y_i = T_i \sin(\delta_i). \quad (3.51)$$

The yaw moment contribution from thruster i is about the vessel's CoG and is obtained from the cross product between position and force:

$$N_i = l_{xi}T_i \sin(\delta_i) - l_{yi}T_i \cos(\delta_i). \quad (3.52)$$

The total generalized force vector is given by the sum of all thruster contributions:

$$\boldsymbol{\tau} = \begin{bmatrix} X \\ Y \\ N \end{bmatrix} = \sum_{i=1}^p \begin{bmatrix} T_i \cos(\delta_i) \\ T_i \sin(\delta_i) \\ l_{xi}T_i \sin(\delta_i) - l_{yi}T_i \cos(\delta_i) \end{bmatrix}. \quad (3.53)$$

This can be compactly written in matrix form as:

$$\boldsymbol{\tau} = \mathbf{B}(\boldsymbol{\delta}) \mathbf{T}, \quad (3.54)$$

where

$$\mathbf{T} = \begin{bmatrix} T_1 \\ \vdots \\ T_p \end{bmatrix}, \quad (3.55)$$

and the control effectiveness matrix is defined as:

$$\mathbf{B}(\boldsymbol{\delta}) = \begin{bmatrix} \cos(\delta_1) & \cdots & \cos(\delta_p) \\ \sin(\delta_1) & \cdots & \sin(\delta_p) \\ l_{x1} \sin(\delta_1) - l_{y1} \cos(\delta_1) & \cdots & l_{xp} \sin(\delta_p) - l_{yp} \cos(\delta_p) \end{bmatrix}. \quad (3.56)$$

The objective of control allocation problem is to minimize the difference between the desired generalized force vector $\boldsymbol{\tau}_d$, provided by the controller (MPC). To account for actuator limitations, this is formulated as a constrained optimization problem with slack variables:

$$\boldsymbol{\tau}_d - h(\mathbf{n}, \boldsymbol{\delta}) = \mathbf{s}, \quad (3.57)$$

where $h(\cdot)$ denotes the nonlinear mapping from actuator input to generalized forces, and $\mathbf{s}$ represents the slack variables that capture the unachievable part of the requested force.

The control vector from the control allocation towards the vessel can be described as:

$$\mathbf{u} = \begin{bmatrix} \mathbf{n} \\ \boldsymbol{\delta} \end{bmatrix}, \quad (3.58)$$

where $\mathbf{n}$ represents the propeller speed and $\boldsymbol{\delta}$ the thruster angles.

The optimization problem is defined by the cost function:

$$J(\mathbf{n}, \boldsymbol{\delta}, \mathbf{s}) = \underbrace{\mathbf{n}^T Q_n \mathbf{n}}_{\text{power penalty}} + \underbrace{(\boldsymbol{\delta} - \boldsymbol{\delta}_{\text{prev}})^T Q_\delta (\boldsymbol{\delta} - \boldsymbol{\delta}_{\text{prev}})}_{\text{rate penalty}} + \underbrace{\mathbf{s}^T Q_s \mathbf{s}}_{\text{force tracking error}}, \quad (3.59)$$

s.t

$$\mathbf{n}_{\min} \leq \mathbf{n} \leq \mathbf{n}_{\max}, \quad (3.60)$$

$$\boldsymbol{\delta}_{\min} \leq \boldsymbol{\delta} \leq \boldsymbol{\delta}_{\max}, \quad (3.61)$$

$$|\delta_i(k) - \delta_i(k-1)| \leq \Delta \delta_{\max}, \quad (3.62)$$

where Q_n , Q_δ , and Q_s are positive semi-definite weighting matrices.

The slack variables $\mathbf{s}$ represent the part of the desired generalized forces that cannot be achieved due to actuator constraints, ensuring feasibility of the optimization problem.

In this work, the control allocation problem is solved using a nonlinear optimization approach rather than an explicit slack-variable formulation.

3.4 CAN bus communication

To enable closed-loop control of a physical marine vessel, communication between the vessel and the controller is required. The CAN bus (Controller Area Network) is a standard serial communication architecture that connects electronic control units (ECUs) throughout a system. The CAN bus reduces the total number of wires needed for communication between the different ECUs. This also contributes to system robustness, as individual modules can fail without affecting the overall system.

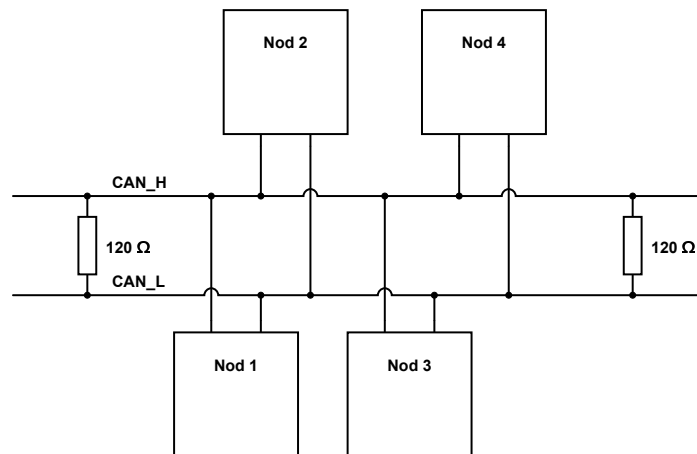


Figure 3.4: A simple illustration of a simple CAN bus.

Each node on the CAN bus is assigned a unique ID used for communication. The identifiers are structured hierarchically, enabling prioritization between different units. All nodes (ECUs) passively listen on the bus and have the ability to both transmit or receive messages.

The hardware consists of a twisted wire pair to reduce electromagnetic interference, as well as terminal resistors to minimize signal reflections. Figure 3.4 illustrates a simplified CAN bus architecture.

The EVC system used in this project communicates via a CAN Bus architecture named MULTILINK, which handles the vessel's internal communication. By using a PEAK PCAN-USB adapter, the control ECU (a computer currently running the

python control script) can communicate with the system over a 250 kbit/s bus. The implementation used the `python-can` library for raw frame handling and `cantools` library for DBC-based signal decoding.

3.4.1 Transmit TX

At each sampling step, the MPC computes normalized control forces, represented in Equation 3.63. These force values are then linearly mapped to match the limits of the EVC system.

$$\mathbf{u} = [u_X, u_Y, u_N] \in [-1, 1] \quad (3.63)$$

,

$$u_{CAN} = u_i * MAX_INPUT \in [MIN_INPUT, MAX_INPUT]. \quad (3.64)$$

The transmit messages are packed into an 8-byte extended CAN frame using little-endian INT16 encoding. To enable real-time operation, including the MPC computations, a dedicated transmit thread broadcasts messages independently of the MPC at 10 Hz. As a safety mechanism, zero-valued frames are transmitted in the event of an MPC failure.

Motor and rudder commands are transmitted using the same approach but are first processed through the control allocator, which maps the desired forces to angles and propeller speeds. These commands are then sent in a separate CAN frame using the same communication structure.

3.4.2 Receive Rx

Vessel state feedback is obtained from a background thread that decodes the incoming message frames and updates the latest measurement.

Table 3.2: CAN bus received signals from the vessel's navigation system.

Signal	Type	Unit	Description
LatitudeWGS84	Position	Degrees	WGS84 latitude from the on-board GNSS receiver.
LongitudeWGS84	Position	Degrees	WGS84 longitude from the on-board GNSS receiver.
LatitudeSpeed	Velocity	m/s	North component of ground-referenced velocity from the GNSS receiver.
LongitudeSpeed	Velocity	m/s	East component of ground-referenced velocity from the GNSS receiver.
HeadingTrue	Direction	Degrees	True heading from the vessel's compass. Used for body-frame velocity rotation and yaw rate estimation.
SpeedOverGround	Velocity	m/s	Scalar speed over ground from the GNSS receiver.
Motorspeed	rotation speed	rpm	Rotations per minute for each motor
Rudder angle	Angle per rudder	Degrees	Current angle of each rudder

The signals were captured on the CAN network where Table 3.2 shows the most important messages for this applications. Measurements of latitudinal and longitudinal coordinates from the onboard GNSS receiver are used to provide the controller with the correct vessel's position. The MPC operates in cartesian coordinates, and therefore a coordinate transformation is required. Docking maneuvers occur within a limited geographical area, where the flat-Earth assumption is considered valid. Under this assumption, the equirectangular approximation can be applied [7]. Alternative methods may yield higher accuracy but incur greater computational cost, while GPS accuracy typically introduces a larger uncertainty. An equirectangular projection centered at the initial received GPS fix (φ_0, λ_0) is used, as defined in Equation 3.65, where M_{lat} represents meters per degree of latitude.

$$\begin{aligned} x_E &= (\lambda - \lambda_0) \cdot M_{lat} \cdot \cos(\varphi_0), \\ y_N &= (\varphi - \varphi_0) \cdot M_{lat}. \end{aligned} \quad (3.65)$$

The velocities received from the measurement are expressed in latitudinal and longitudinal units and must be transformed into the vessel's body-fixed frame. This transformation is performed using the NED speed components in combination with the measured heading ψ , as shown in Equation 3.66 and Equation 3.67.

$$u_{body} = V_N \cos(\psi) + V_E \sin(\psi), \quad (3.66)$$

$$v_{body} = -V_N \sin(\psi) + V_E \cos(\psi). \quad (3.67)$$

The yaw angle of the marine vessel corresponds to the measured heading, and the yaw rate r is estimated from successive heading measurements as shown in Equation 3.68.

$$r = \frac{\text{atan2}(\sin(\psi_k - \psi_{k-1}), \cos(\psi_k - \psi_{k-1}))}{\Delta t}. \quad (3.68)$$

4

Results

This chapter presents the results obtained from simulations and experimental evaluations conducted to assess the performance of the proposed vessel control framework. The results include validation of the dynamic vessel model, autonomous docking performed in Python and on a simulation rig, and experimental sea trials.

The performance of the calibrated dynamic model, the MPC, the thrust allocation algorithm, and the EKF is evaluated both individually and as an integrated system. Model validation is carried out by comparing simulated marine vessel responses with experimental data, while the control and estimation components are based on tracking accuracy, robustness, and constraint handling during navigation and docking maneuvers.

4.1 Model validation

The dynamic vessel model evaluated in this work is the MAM. The goal of these experiments is to obtain a set of model parameters that minimizes the discrepancy between the estimated model response and the simulated output data. The baseline for this validation is the simulated vessel from CPAC's boat simulation rig.

4.1.1 Calibration on simulation rig

As the first part of the calibration procedure, step response experiments were conducted on the simulation rig (see Figure 4.1) to isolate the three main motion components.

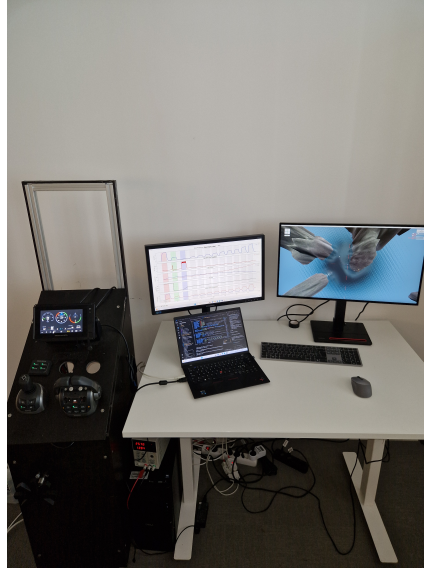


Figure 4.1: Simulation rig setup.

The resulting step responses are shown in Figure 4.2. These experiments were used to excite and isolate the three primary motion components, surge, sway, and yaw. This allows the corresponding dynamic behavior to be identified in each degree of freedom.

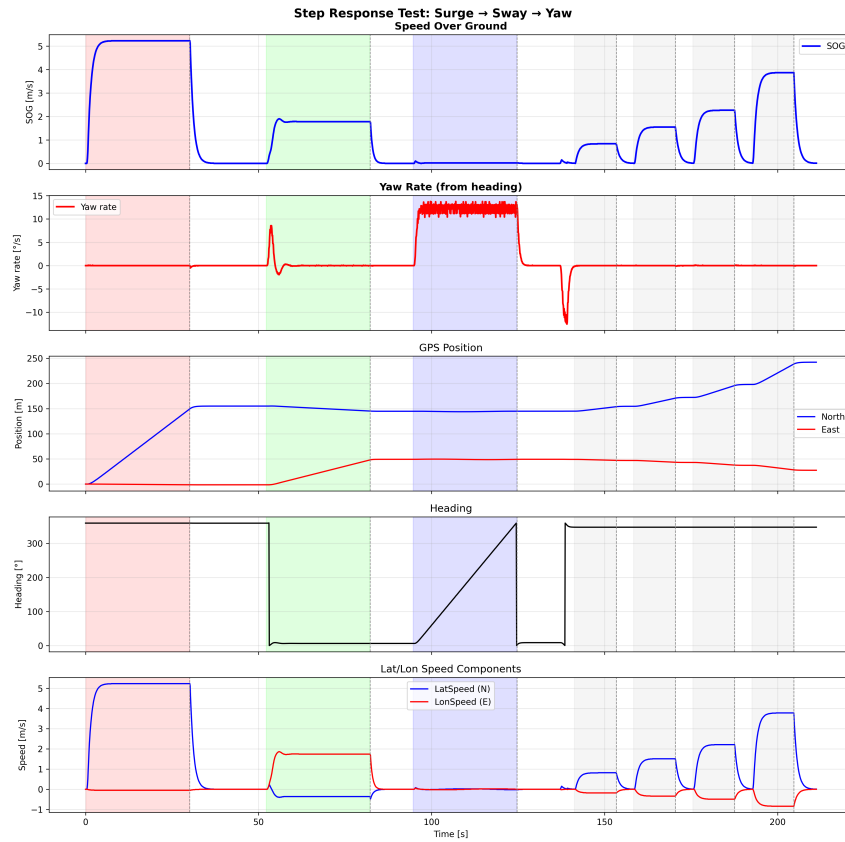


Figure 4.2: Logged step response conducted on the simulation rig.

As shown in Figure 4.2, the operation limits of the system can be observed from the calibration experiments, including vessel speed, yaw rate, and positioning. Based on these identified step responses, the operational bounds of the system were determined. These limits include velocity, force, and effective mass parameters derived from both acceleration and coast-down experiments. The resulting calibrated parameters are summarized in Table 4.1.

Table 4.1: Parameters derived from vessel step-response calibration (updated).

Category	Parameter	Value	Derivation
Velocity limits	$u_{\max}$	$\pm 5.32 \text{ m/s}$	Measured max surge SOG
	$v_{\max}$	$\pm 1.90 \text{ m/s}$	Measured max sway SOG
	$r_{\max}$	$\pm 13.75 \text{ deg/s}$	Measured max yaw rate
	$n_{\max}$	458.8 RPM	Steady-state surge RPM
Force limits	$X_{\max}$	14 317.0 N	$2 T_1^{\text{surge}}$ (459 RPM)
	$Y_{\max}$	10 392.5 N	$2 T_1^{\text{sway}} \sin \delta_{\max}$ (780 RPM)
	$N_{\max}$	9 370.0 Nm	$2 l_{y_t} k_1 \text{ RPM}_{yaw}$ (610 RPM)
Effective mass (acceleration)	m_{11}^{acc}	14 352.9 kg	$X_{\max}/a_0^u$, $a_0^u = 0.9975 \text{ m/s}^2$
	m_{22}^{acc}	14 869.8 kg	$Y_{\max}/a_0^v$, $a_0^v = 0.6989 \text{ m/s}^2$
	m_{33}^{acc}	109 309.3 kg m ²	$N_{\max}/\dot{r}_0$, $\dot{r}_0 = 0.085720 \text{ rad/s}^2$
Effective mass (coast-down)	m_{11}^{cst}	16 448.0 kg	$X_{\max}/ d^u $, $ d^u = 0.8704 \text{ m/s}^2$
	m_{22}^{cst}	19 649.3 kg	$Y_{\max}/ d^v $, $ d^v = 0.5289 \text{ m/s}^2$
	m_{33}^{cst}	148 382.1 kg m ²	$N_{\max}/ d^r $, $ d^r = 0.063148 \text{ rad/s}^2$
Combined mass	m_{11}	15 400.5 kg	$0.5 m_{11}^{\text{acc}} + 0.5 m_{11}^{\text{cst}}$
	m_{22}	17 259.5 kg	$0.5 m_{22}^{\text{acc}} + 0.5 m_{22}^{\text{cst}}$
	m_{33}	128 845.7 kg m ²	$0.5 m_{33}^{\text{acc}} + 0.5 m_{33}^{\text{cst}}$

The following step in the identification sequence is the trajectory-based calibration approach, which was applied to estimate the remaining parameters. This yields a zig-zag trajectory, which is shown in subsection 3.1.2, that is used as a reference input for the GA. Figure 4.3 illustrates the key states monitored during this calibration phase.

4. Results

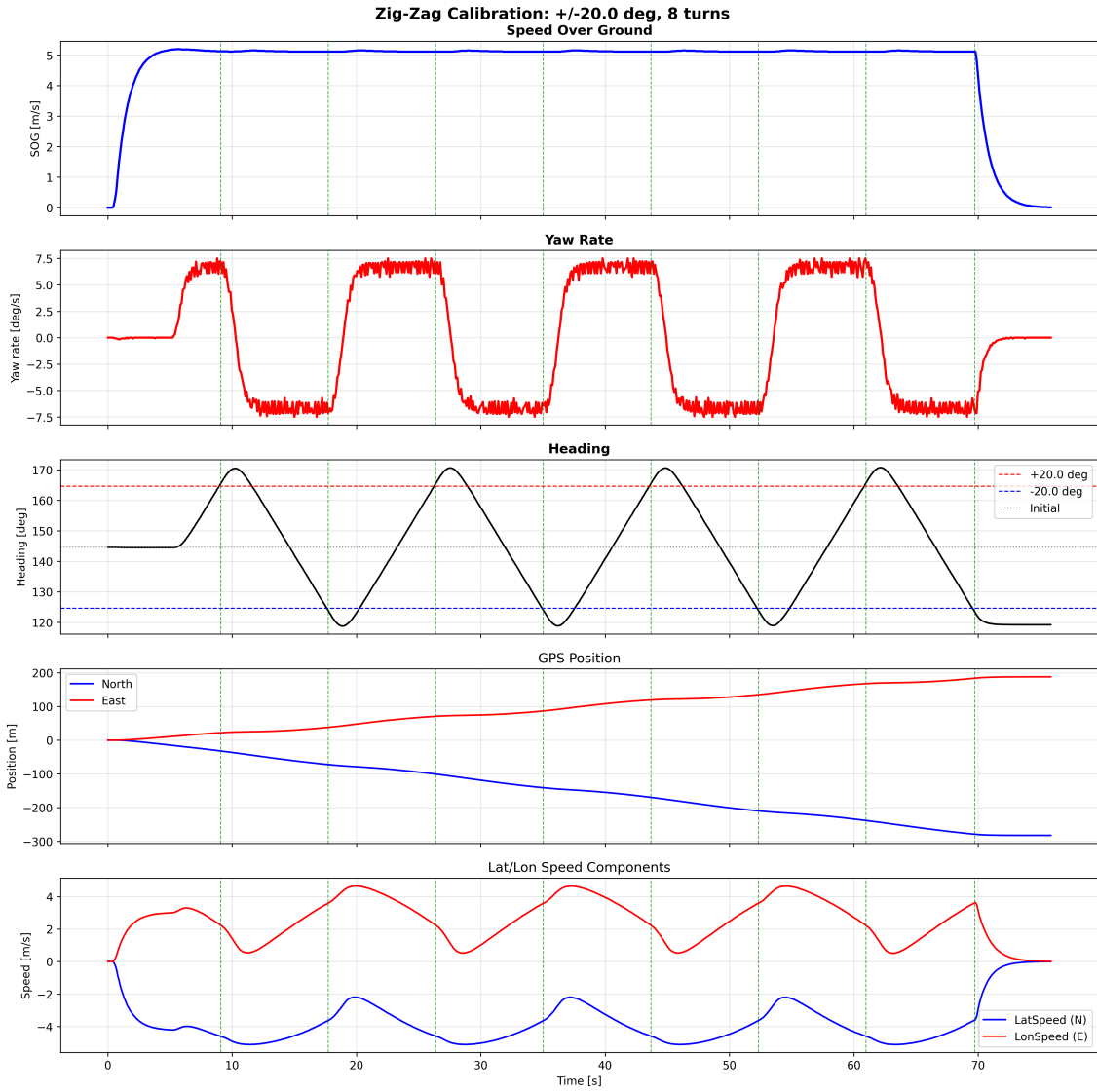


Figure 4.3: 2020-zigzag calibration plot of important states.

The subsequent calibration step uses the zigzag path described in subsection 3.1.2 to estimate the system's hydrodynamic parameters. The resulting state estimates are compared against measured data, with the fitted trajectories shown in Figure 4.4 and Figure 4.5. The generated values of the hydrodynamics can be seen in Table 4.2.

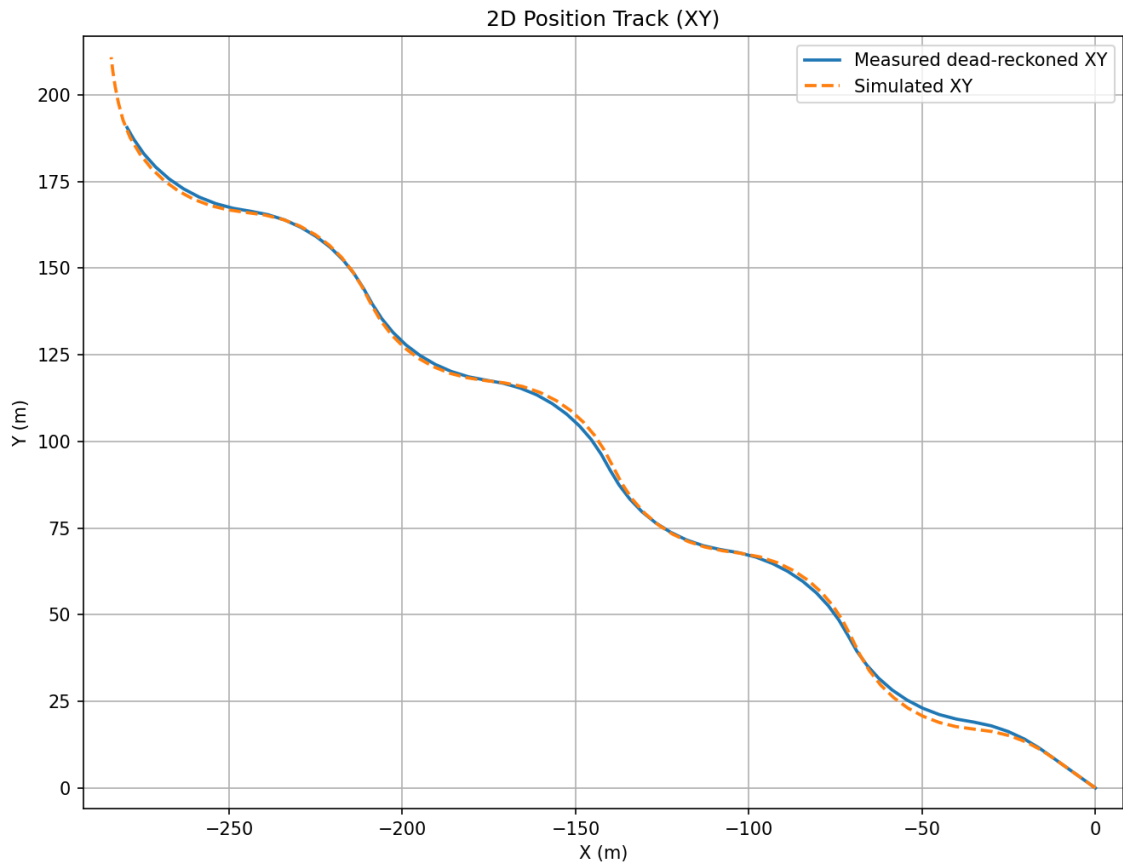


Figure 4.4: The position tracking from the GA-Algorithm

The genetic algorithm (GA) minimizes a weighted sum of tracking errors across selected states Figure 4.6. Higher weights, see Table 4.3, are assigned to surge velocity and heading due to their dominant role in docking control, while positional errors are given lower priority during identification.

4. Results

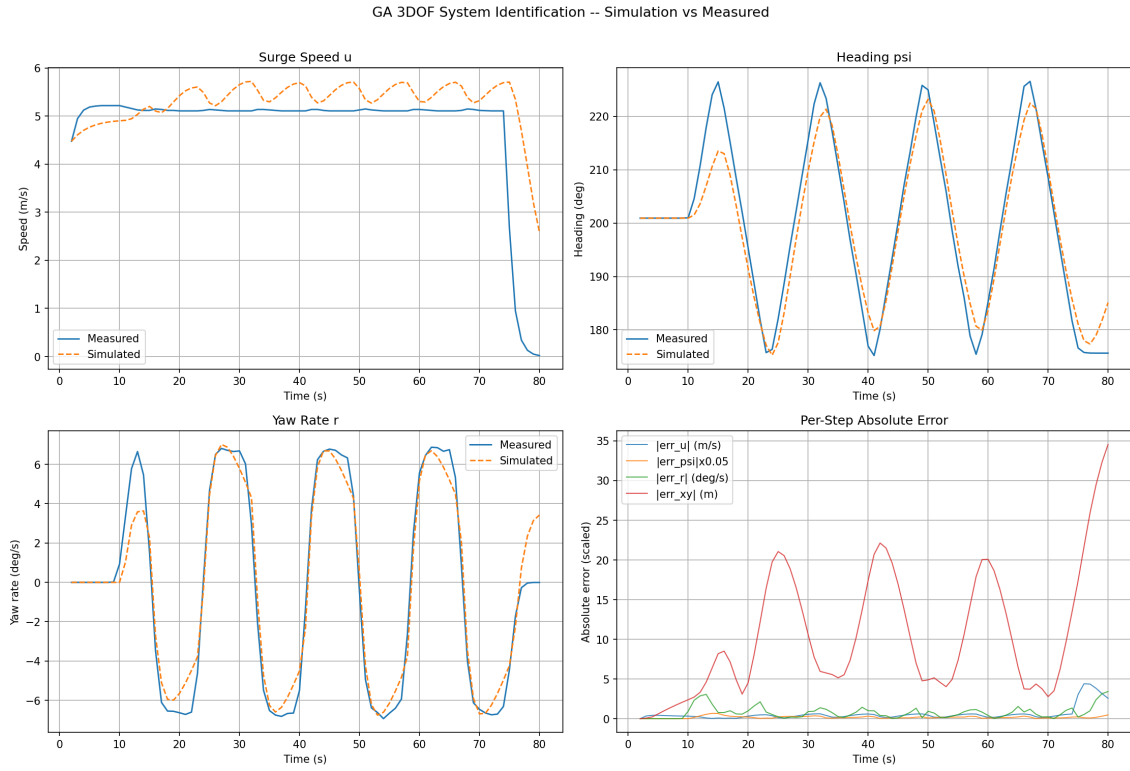


Figure 4.5: The state tracking from the GA-Algorithm.

Table 4.2: Identified Abkowitz maneuvering model parameters (GA-updated).

Category	Parameter	Value
<i>Surge hydrodynamic derivatives</i>		
X_u	Linear surge damping	-1998.01
X_{uu}	Quadratic surge damping	-347.05
<i>Sway hydrodynamic derivatives</i>		
Y_v	Linear sway damping	-7365.78
Y_r	Yaw-induced sway force	42 937.56
<i>Yaw hydrodynamic derivatives</i>		
N_r	Linear yaw damping	-20 426.43
N_{rr}	Quadratic yaw damping	-42 591.68
N_v	Sway-induced yaw moment	-2 220.72
<i>Actuation-related parameters</i>		
k_1	Thrust gain coefficient	15.08
k_2	Nonlinear thrust coefficient	492.37
Y_d	Differential thrust sway gain	592.81
N_d	Differential thrust yaw gain	143.40
l_{yt}	Lateral thruster lever arm	0.50 m

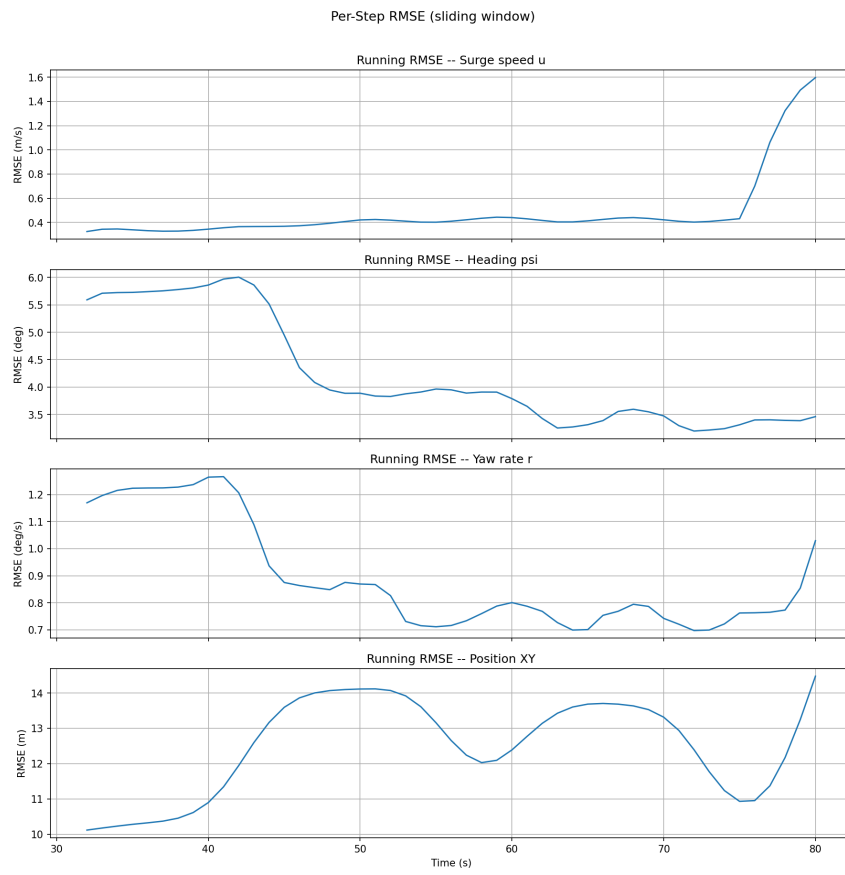


Figure 4.6: RMSE of important GA identified states.

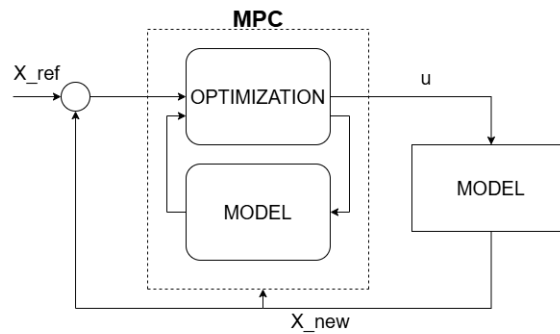
Table 4.3: Weighting factors used in GA-based parameter identification.

State	Weight
W_u	20.0
W_ψ	10.0
W_r	5.0
W_x	1.0
W_y	1.0

4.2 Validation of MPC

The MPC performance is evaluated in a Python environment using a preliminary tuning. The plant model matches the MPC model to isolate controller performance from model mismatch. This provides MPC evaluation results without any other disturbance or model mismatch. In Figure 4.7, an illustration of the architecture can be seen.

The MPC successfully drives the system to the terminal states with good trajectory tracking as shown in Figure 4.8 and Figure 4.10. In Figure 4.9 and Figure 4.11 there are two different references. One representing the penultimate and terminal state while the other represents the local horizon reference. The last part of the trajectory between the penultimate and terminal state is a separate MPC tuning called docking mode. The docking mode penalizes heading deviations and drives the system to the terminal state.

**Figure 4.7:** MPC controller architecture illustration.

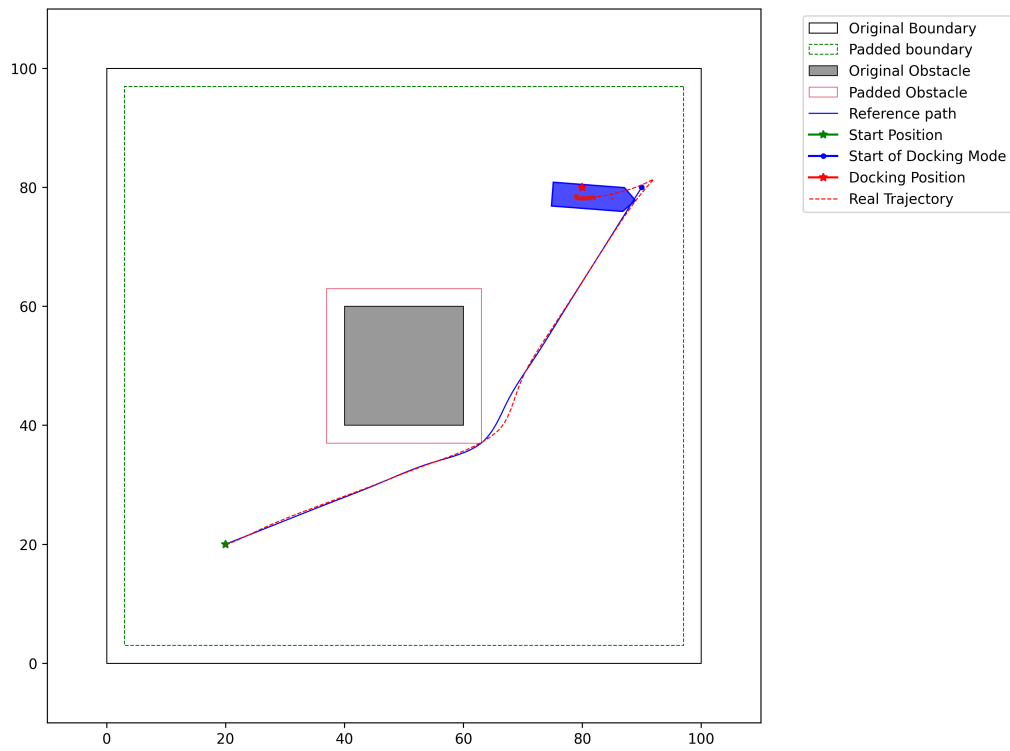


Figure 4.8: MPC validation using python models as plant.

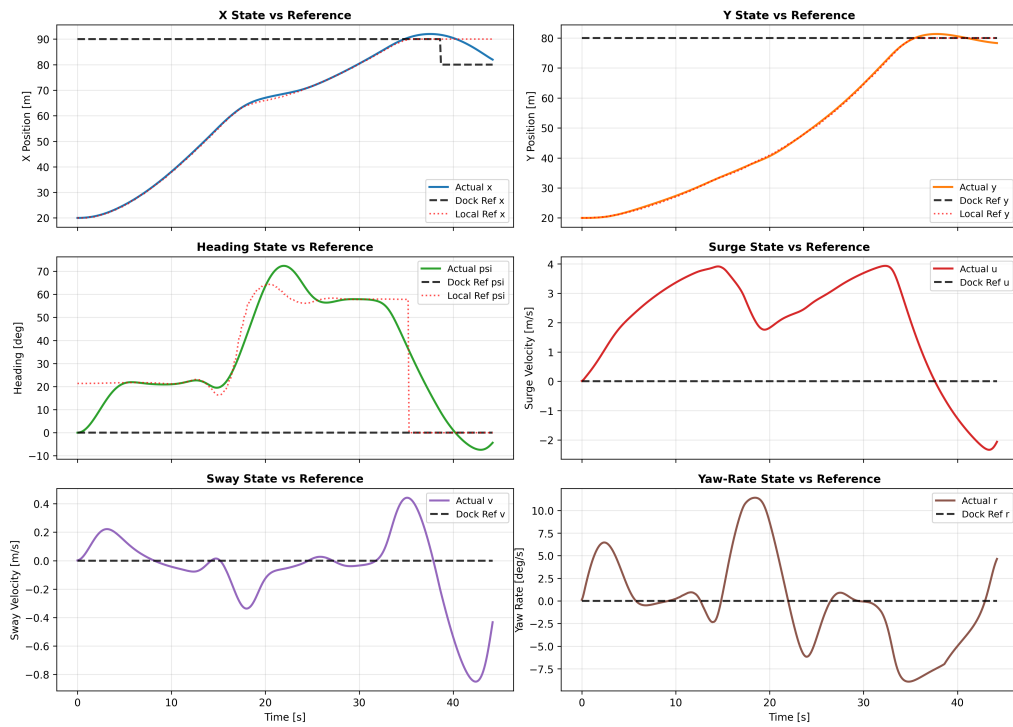


Figure 4.9: States of MPC validation using python models as plant.

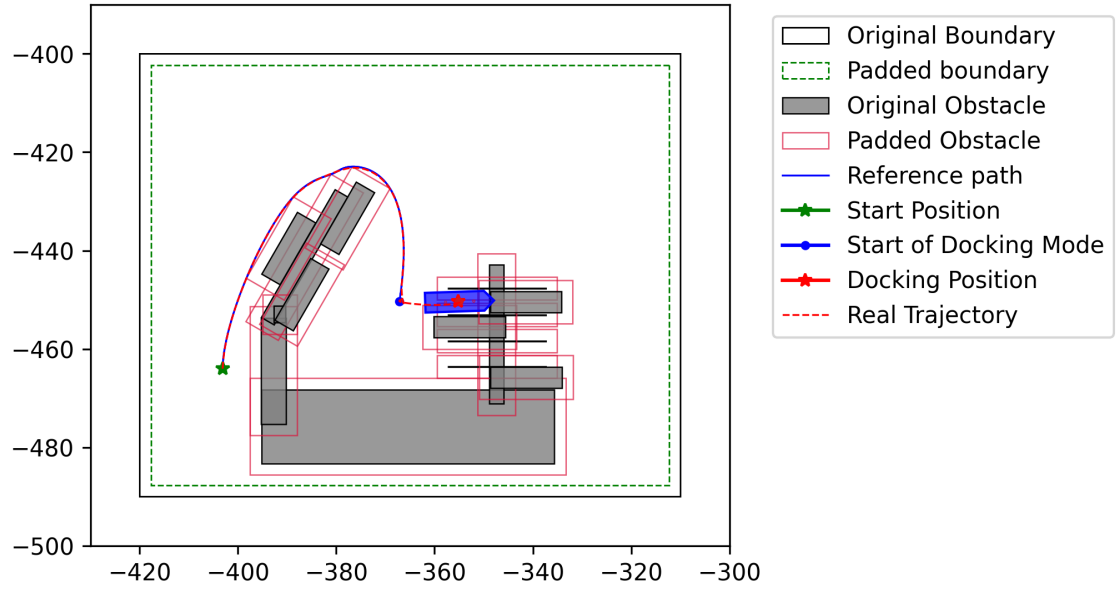


Figure 4.10: MPC validation using python models as plant.

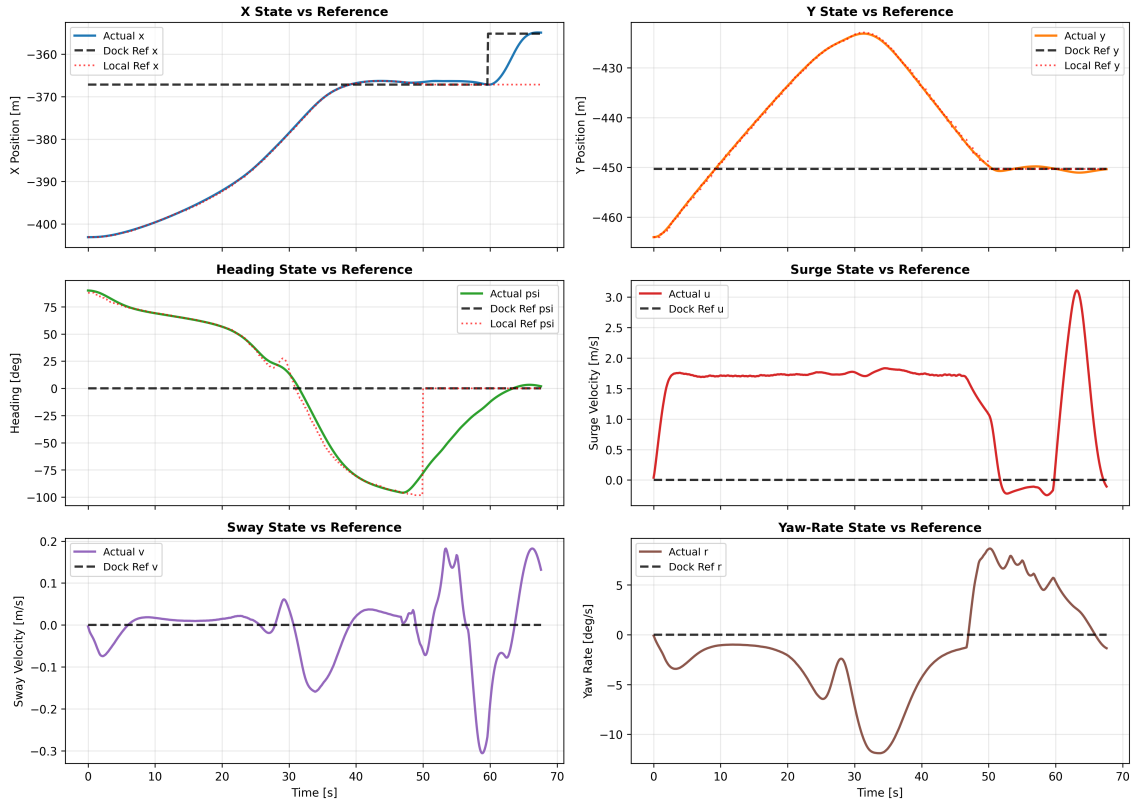


Figure 4.11: States of MPC validation using python models as plant.

4.3 Validation of Extended Kalman Filter

The EKF was built upon the existing system model and was extended to estimate the hidden states of the disturbances b_u, b_v, b_r . The system state vector consists of

position vector $\eta = [x, y, \psi]^T$, the velocity vector $\nu = [u, v, r]^T$, and the disturbance vector $b = [b_u, b_v, b_r]^T$. The disturbance states are modeled as an equivalent body-frame acceleration per unit mass. The measured states are set as the position and heading, and the full state vector is given as

$$x = \begin{bmatrix} \eta \\ \nu \\ b \end{bmatrix} = \begin{bmatrix} x_0 \\ y_0 \\ \psi \\ u \\ v \\ r \\ b_u \\ b_v \\ b_r \end{bmatrix}. \quad (4.1)$$

The process variance for position is set to $\mathbf{q}_p = 0.001$. The process variance for velocity is set to $\mathbf{q}_v = 0.0005$. The bias process noise is set to $\mathbf{R}_p = 1e-6$. The measurement noise covariance is $\mathbf{R}_p = 0.25$ and $\mathbf{R}_\theta = 0.0025$.

To verify the EKF functionality, a simulation of the system was made. Starting from a steady state $X_0 = [0, 0, 0, 0, 0, 0, 0, 0, 0]^T$, white Gaussian noise (WGN) was applied at each simulation step to simulate measurement noise, with a standard deviation of 0.5^2 with no other disturbances present. The results of this static validation are shown in Figure 4.12 and Figure 4.13. These figures demonstrate that the EKF effectively reduces measurement noise and provides stable disturbance estimates.

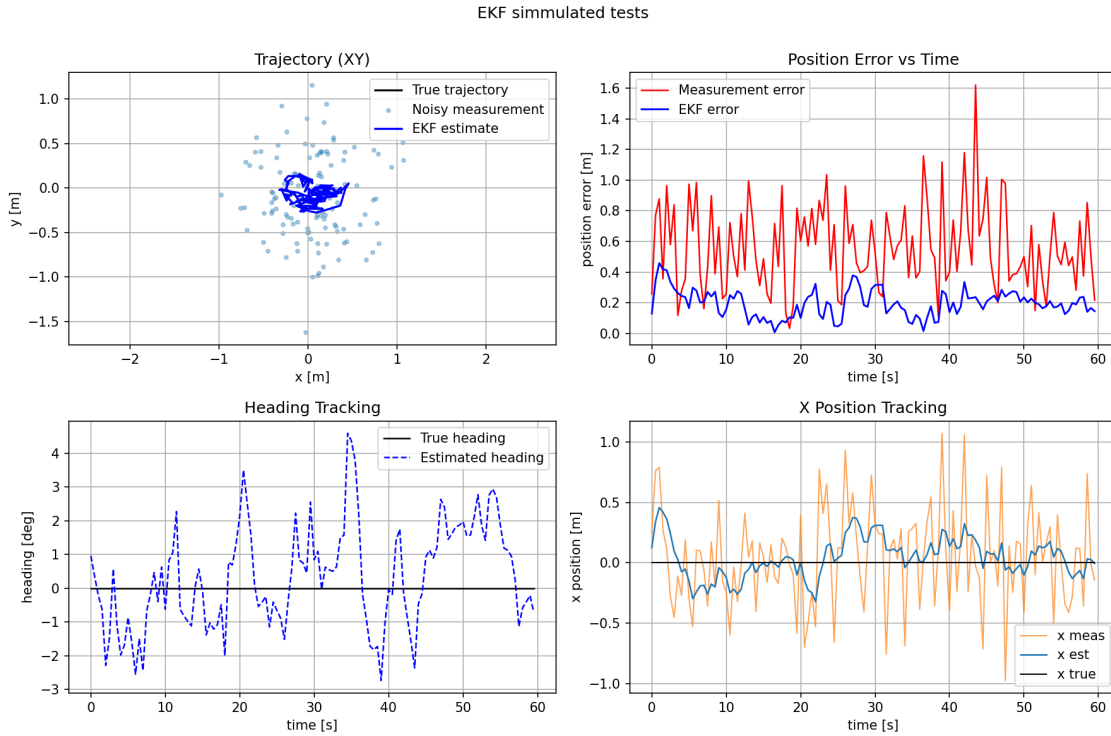


Figure 4.12: Plot of a static EKF Simulation

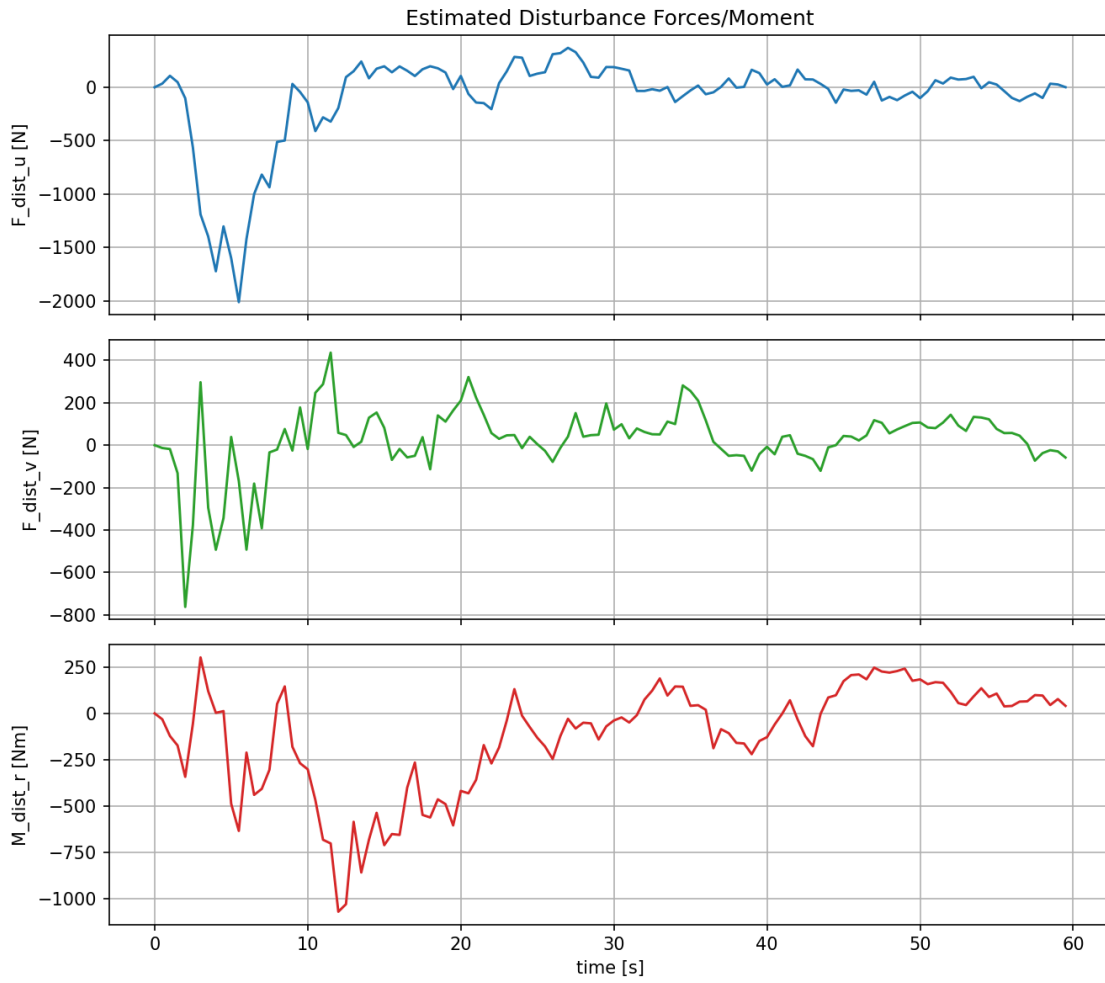


Figure 4.13: Plot of a static EKF Simulation, disturbance estimates.

In order to evaluate the EKF under dynamic conditions, a constant surge input was applied. The response of the estimator is shown in Figure 4.14 and Figure 4.15. These results demonstrate that the EKF maintains accurate state estimation under motion.

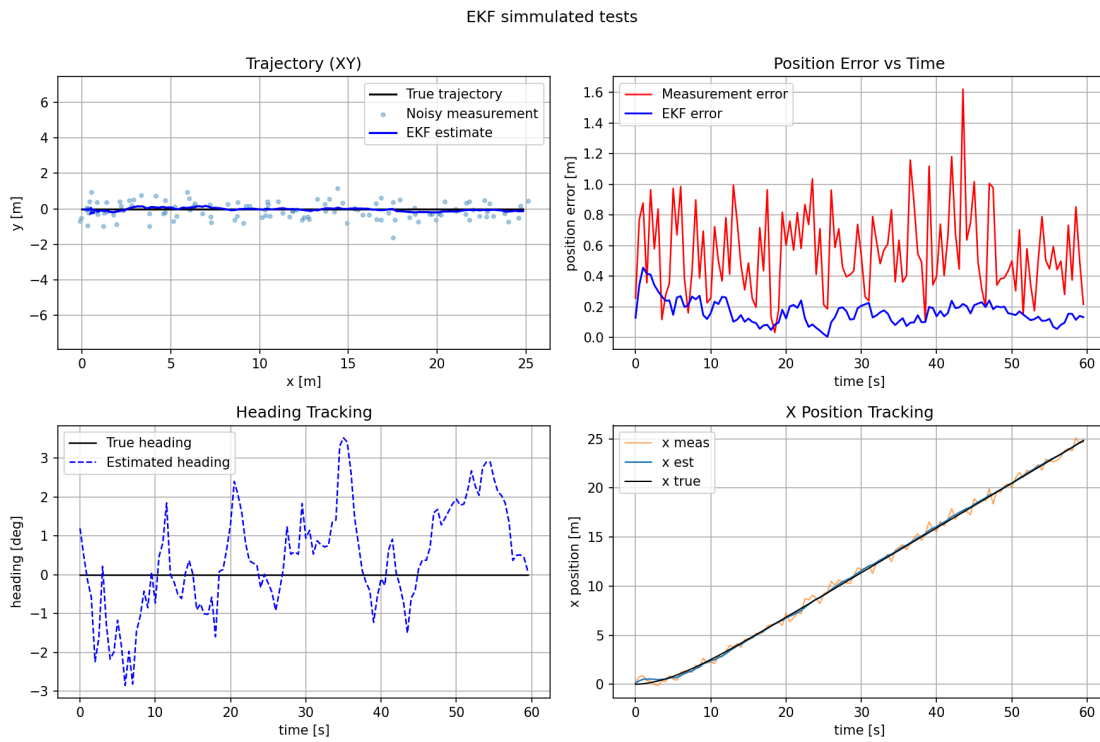


Figure 4.14: Plot of EKF Simulation, constant surge force.

A constant disturbance force of 500 N was applied in the surge direction to verify the estimation of the biased states. The results in Figure 4.16 and Figure 4.17 show that the EKF is capable of identifying and estimating the introduced disturbance.

4. Results

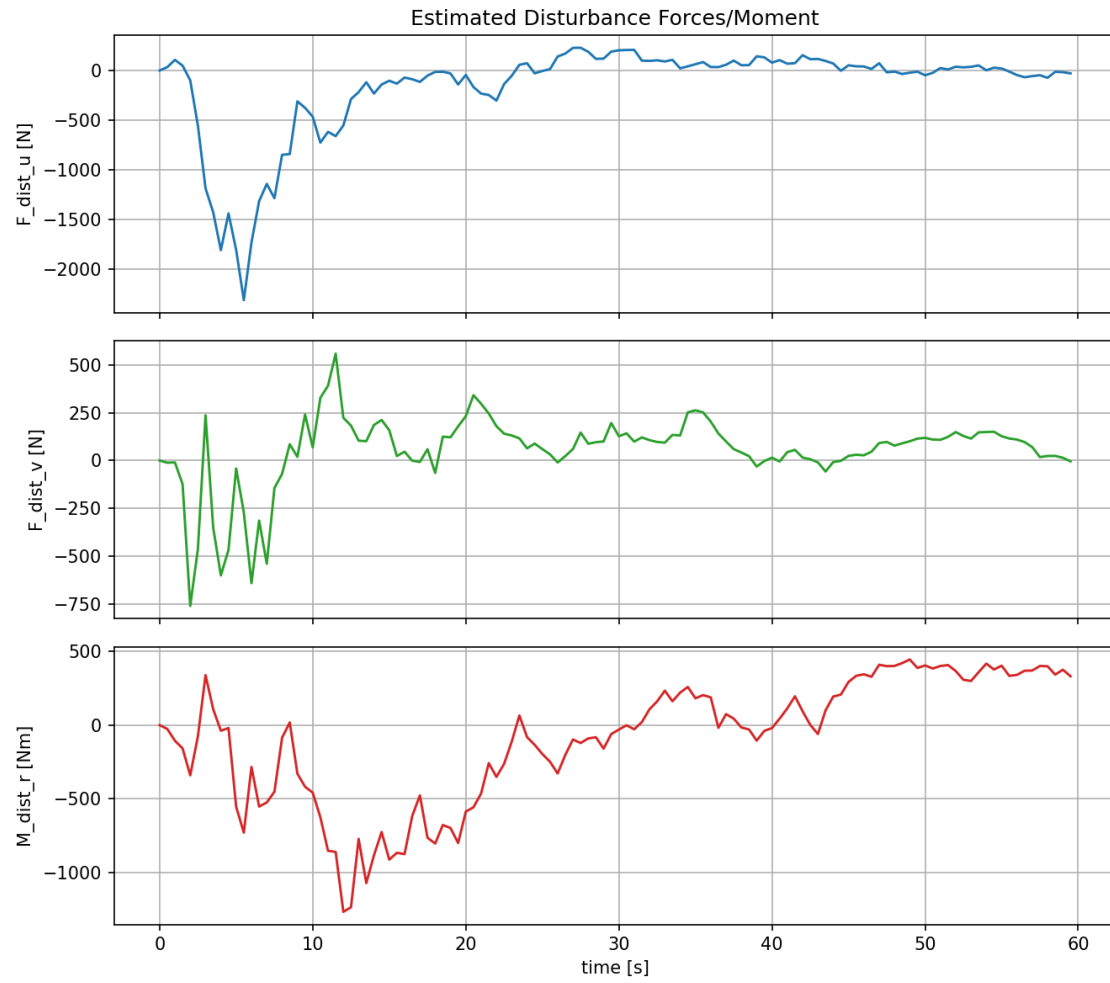


Figure 4.15: Plot of EKF Simulation, no added bias.

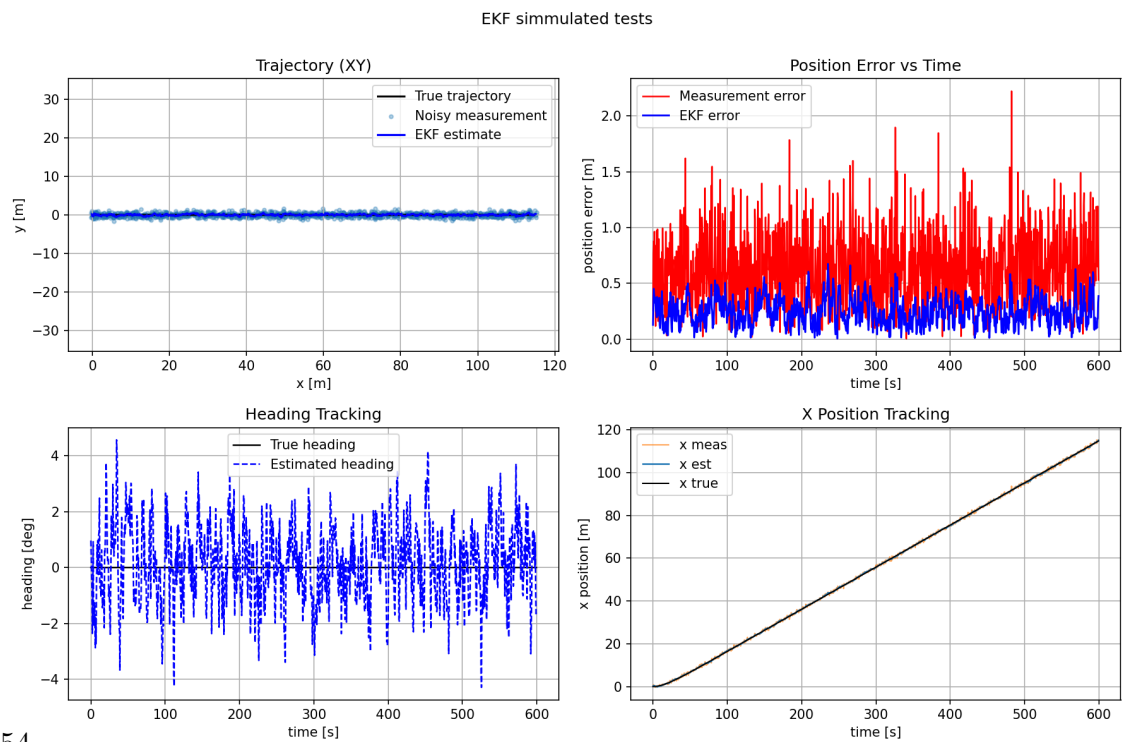


Figure 4.16: Plot of EKF Simulation, 500 N added bias in surge

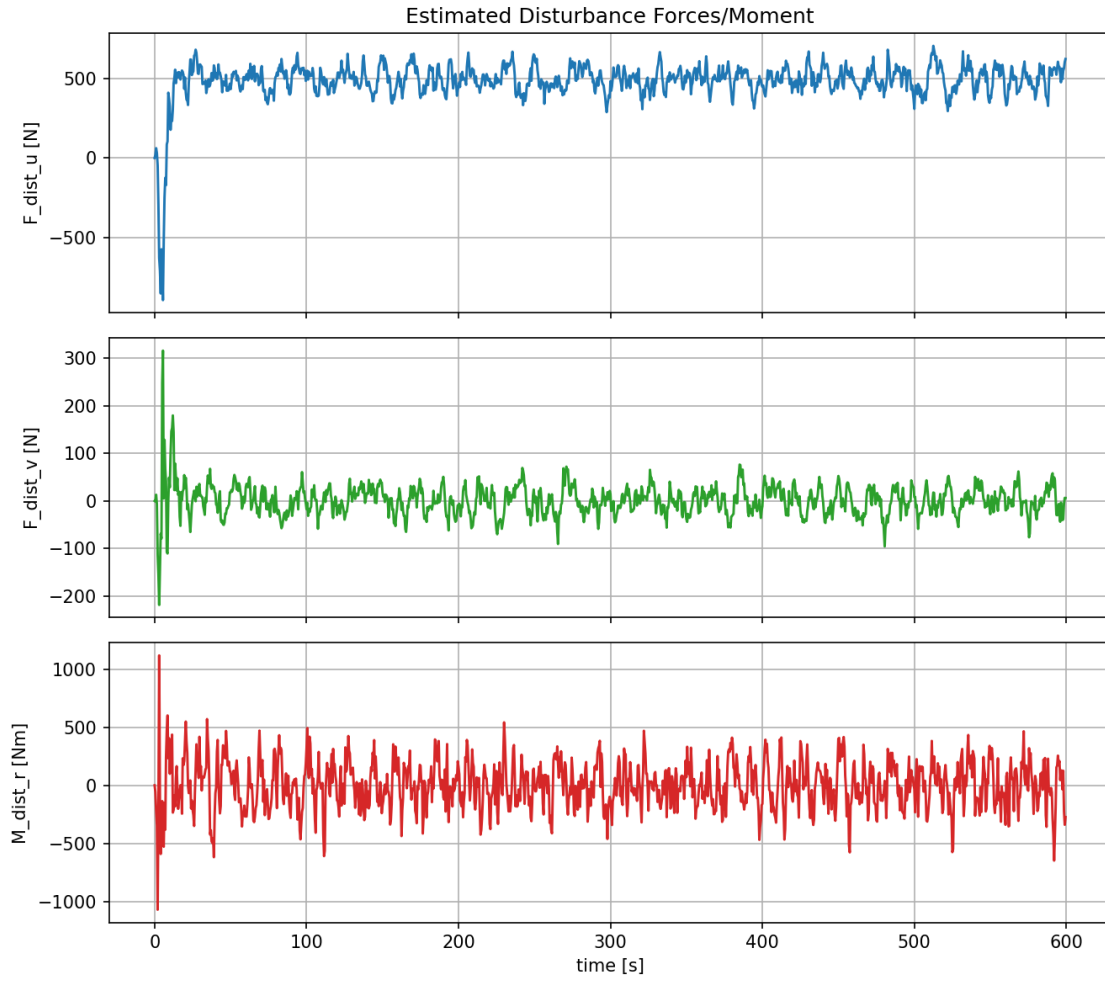


Figure 4.17: Disturbance estimate plot of EKF Simulation, 500 N added bias in surge.

4.3.1 Acceleration vs IMU

To reduce sensor count and hardware complexity, acceleration is estimated from existing sensor data. For this system the acceleration is derived from the velocity measurements provided by a GPS system, supplied as a necessity for the autonomous docking system. Consequently, several coast-down tests were conducted using both high-precision GPS and IMU data to compare their acceleration estimation accuracy, with the IMU serving as the reference. Two CAN-data logs were merged in time and the acceleration was processed and filtered. The acceleration deviation was calculated as in Equation 4.2.

$$\hat{a} = \frac{\Delta V_{surge}}{\Delta t} \quad (4.2)$$

The data was later filtered using a Moving average filter. The IMU measurements were filtered using a low-pass filter. The test was made from a coast-down maneuver, purely surge maneuver.

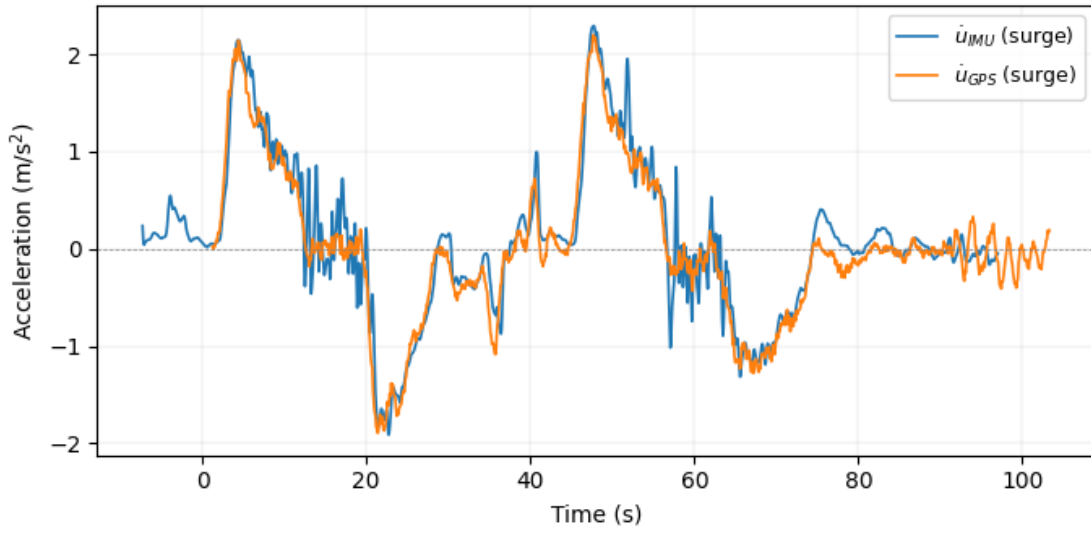


Figure 4.18: Acceleration derived from GPS-velocity vs IMU data.

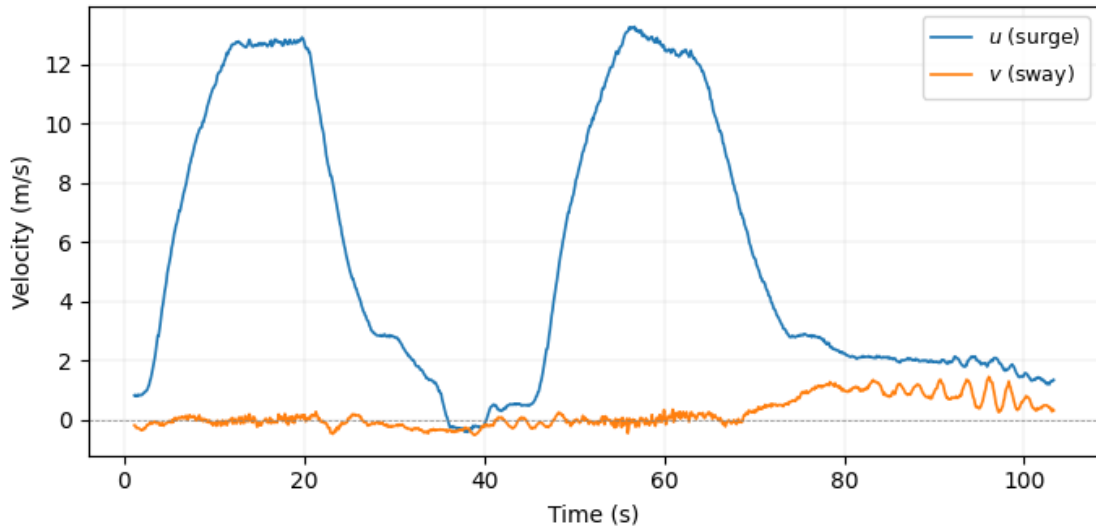


Figure 4.19: Velocity plot for two coast-down maneuvers.

As shown in Figure 4.19, the slope of the velocity during the surge coast-down maneuver forms the basis for the acceleration estimation. Furthermore, Figure 4.18 demonstrates that the GPS-based acceleration estimate is sufficiently close to the IMU measurements, indicating that no additional IMU sensor is required for this application.

4.4 Comparison of the complete system

The full pipeline was implemented, starting with system identification of the model followed by control using a basic MPC tuning. The control signals were transmitted to the plant, where the measurements were processed using an EKF filter. The final

docking objective is defined relative to a penultimate state, where the terminal state is introduced as a small positional and heading offset to ensure a smooth transition into the docking configuration. For the simulation rig experiments, convergence to the terminal state is considered successful when the vessel reaches a positional error within 0.5 m and a heading error within $\pm 5^\circ$.

4.4.1 Scenario 1 on simulation rig

From an initial calibration procedure from subsection 4.1.1, the entire system was tested on the simulation rig. Following calibration and system identification on the simulation rig, the MPC-controlled vessel successfully reached the terminal state illustrated in Figure 4.20. The corresponding error evaluation related to the reference can be observed in Figure 4.21.

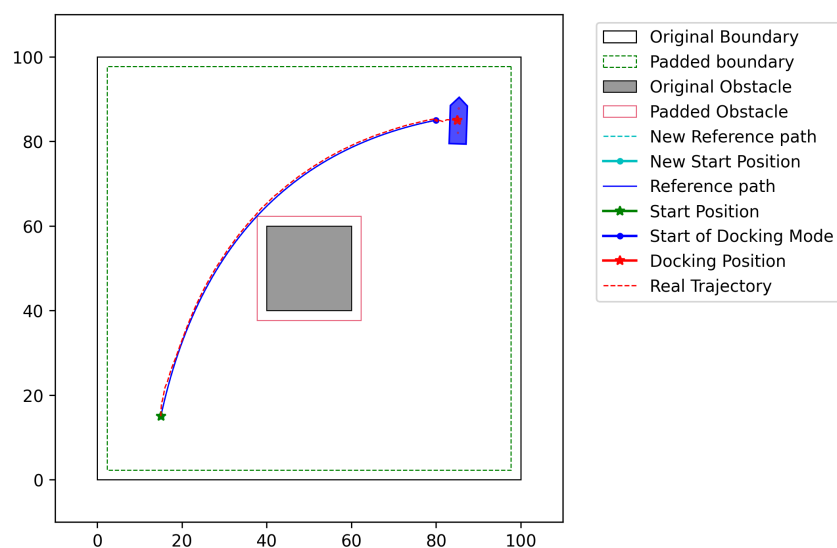


Figure 4.20: Simple trajectory following.

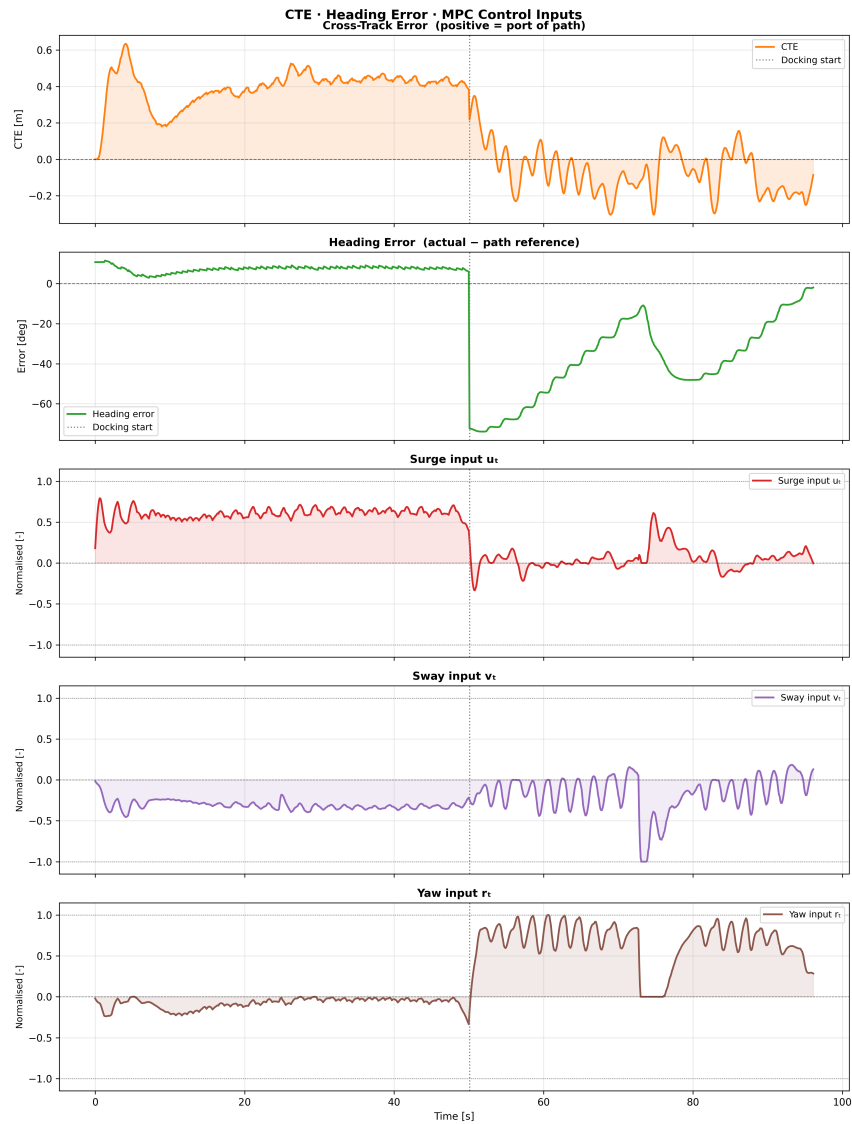


Figure 4.21: Plots of CTE, heading error and the corresponding control inputs.

4.4.2 Scenario 2 on simulation rig

A second scenario, as shown in Figure 4.22, features more demanding turns. This was validated using the same calibration and tuning seen in Table 4.5. The system exhibited tracking degradation during sharp turns near obstacles. The behavior is further illustrated in Figure 4.23.

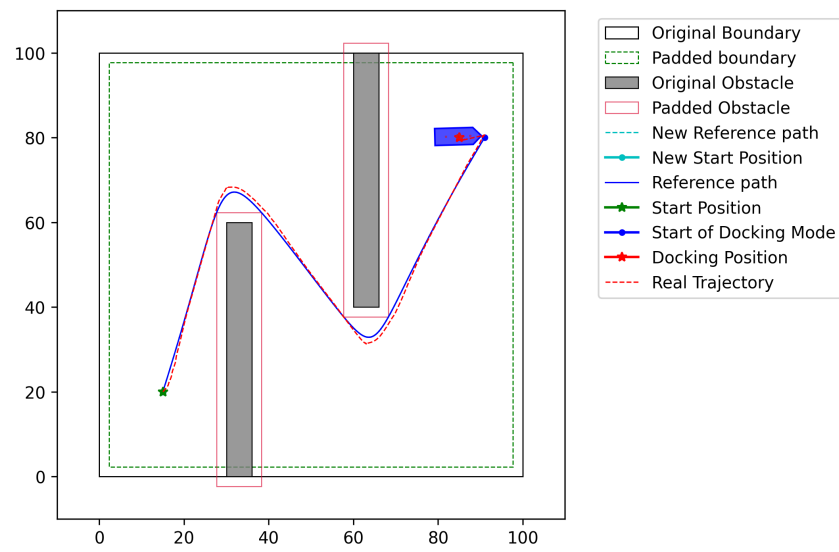


Figure 4.22: Trajectory following.



Figure 4.23: Plots of CTE, heading error and the corresponding control inputs.

4.4.3 Scenario 3 on simulation rig

Scenario 3 handles greater complexity, as shown in Figure 4.24, where both the maneuvering and the docking procedure are more demanding. The corresponding state tracking performance is shown in Figure 4.25

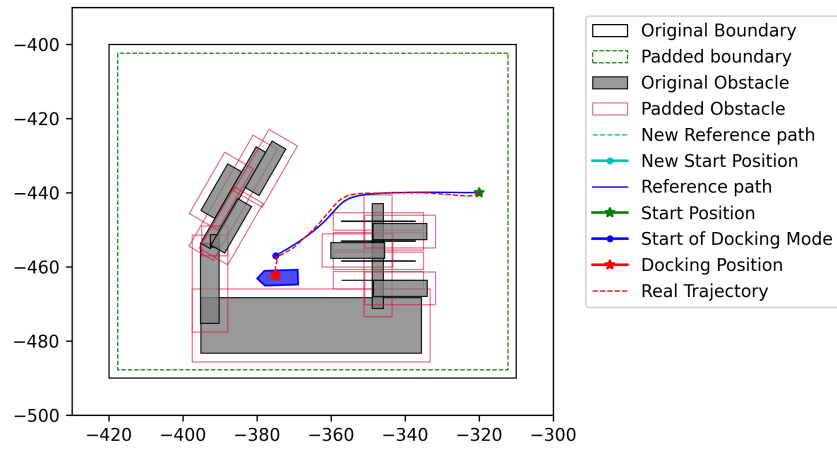


Figure 4.24: Scenario 3: Obstacle avoidance with tight docking.

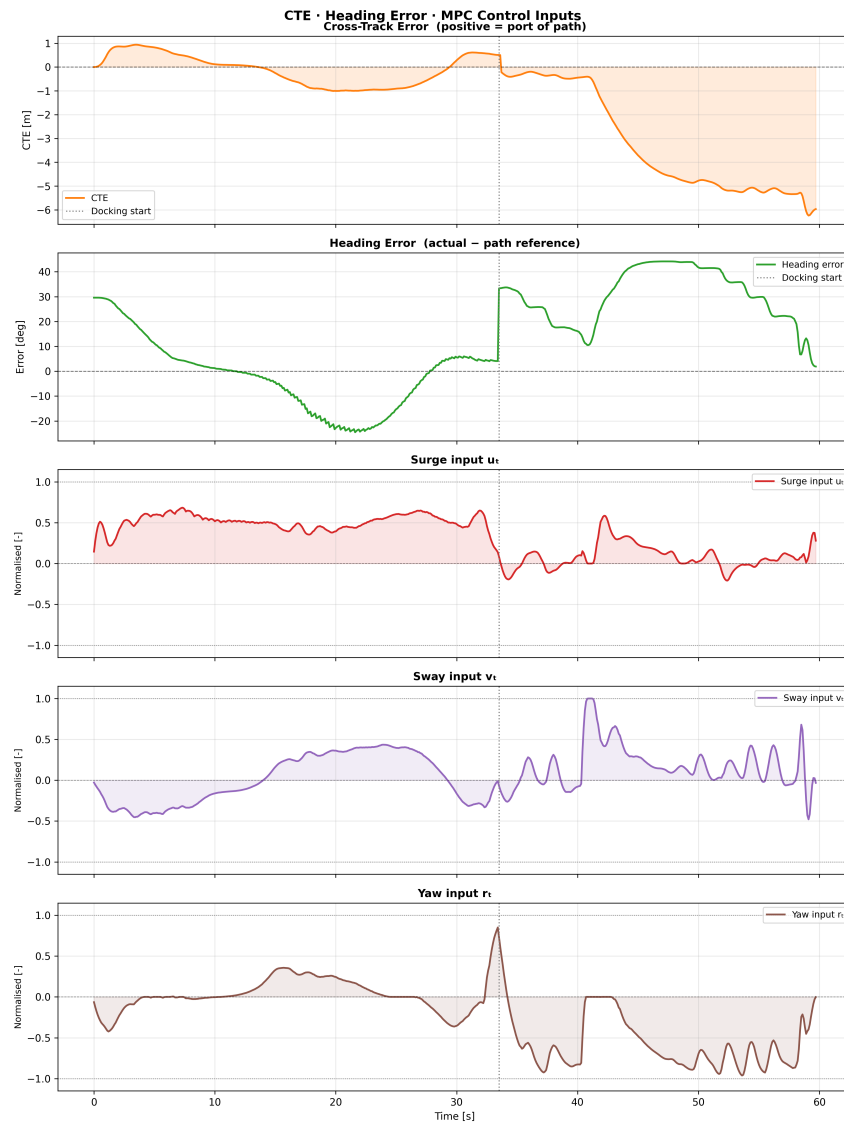


Figure 4.25: Plots of CTE, heading error and the corresponding control inputs.

A second configuration of Scenario 3 is presented in Figure 4.26, where different initial and terminal position are considered. The vessel successfully adapts to the new setup, as seen in the trajectory plot, while the corresponding state tracking is shown in Figure 4.27.

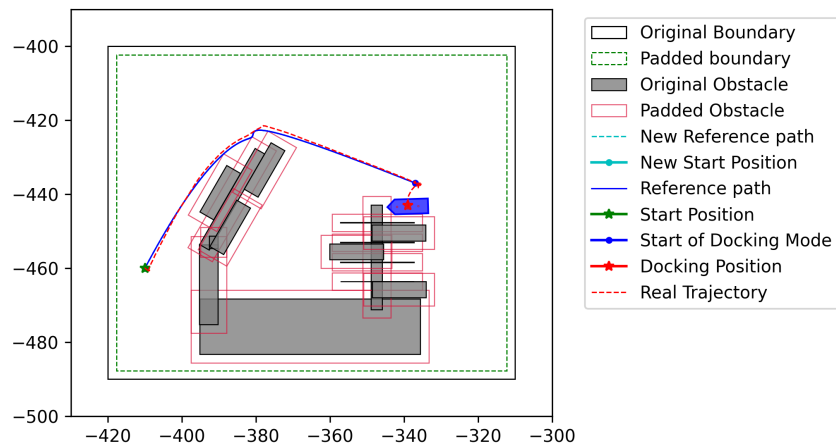


Figure 4.26: Scenario 3.2: Obstacle avoidance with tight docking.



Figure 4.27: Plots of CTE, heading error and the corresponding control inputs.

A third more complex configuration was tested Figure 4.28 and the errors were logged Figure 4.29

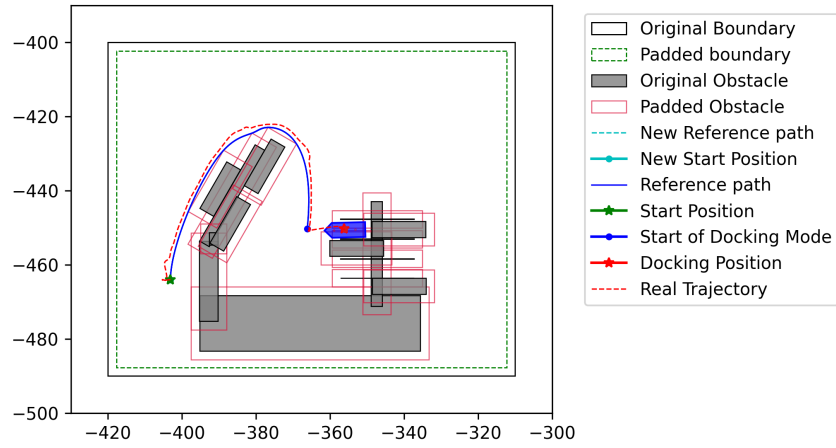


Figure 4.28: Scenario 3.3: Obstacle avoidance with tight docking.

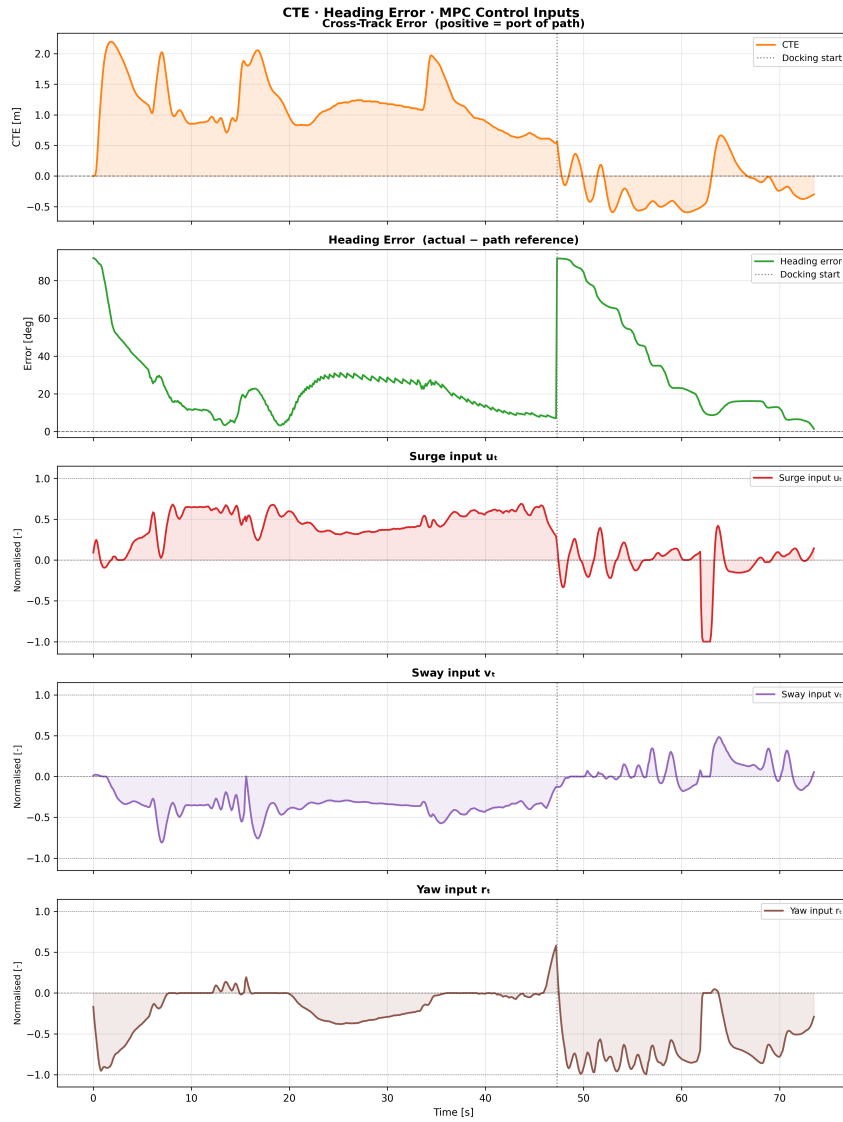


Figure 4.29: Plots of CTE, heading error and the corresponding control inputs.

4.4.4 Extended Kalman filter compensation

Finally, disturbances representing wind, waves, and current were added in the simulation in addition to model imperfections. This was done to evaluate the system's disturbance rejection capabilities.

The resulting vessel trajectory is shown in Figure 4.30, where the system successfully reaches the terminal state despite the disturbances. The corresponding CTE, heading error, and control inputs are presented in Figure 4.31.

The EKF-estimated disturbance does not fully converge to a constant disturbance as intended, see Figure 4.32.

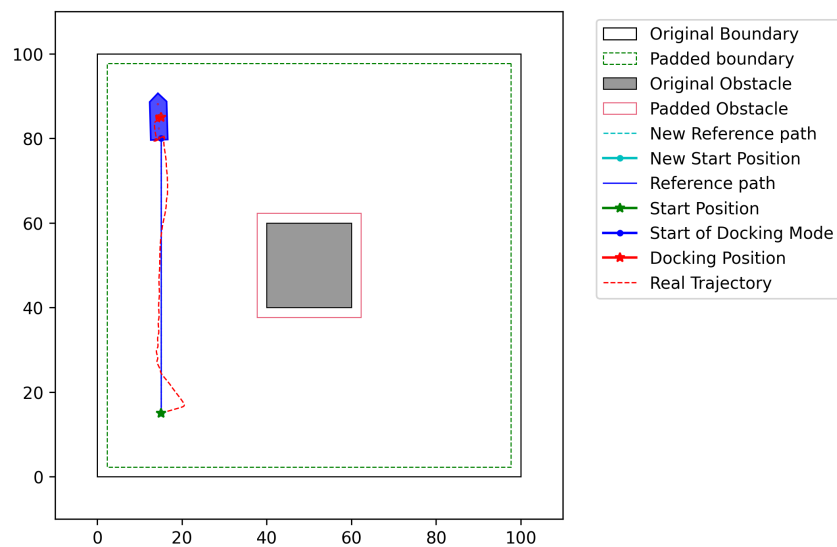


Figure 4.30: Disturbance handling trajectory.

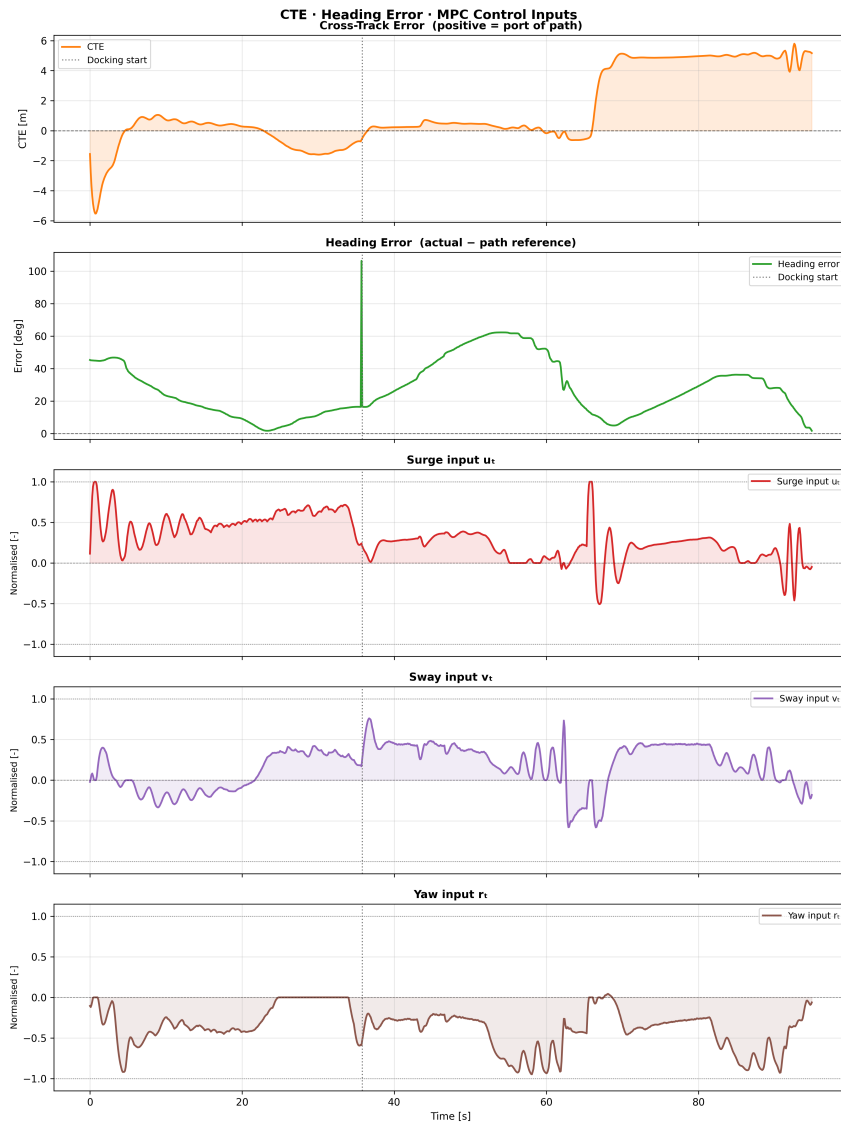


Figure 4.31: Plots of CTE, heading error and the corresponding control inputs for disturbance handling.

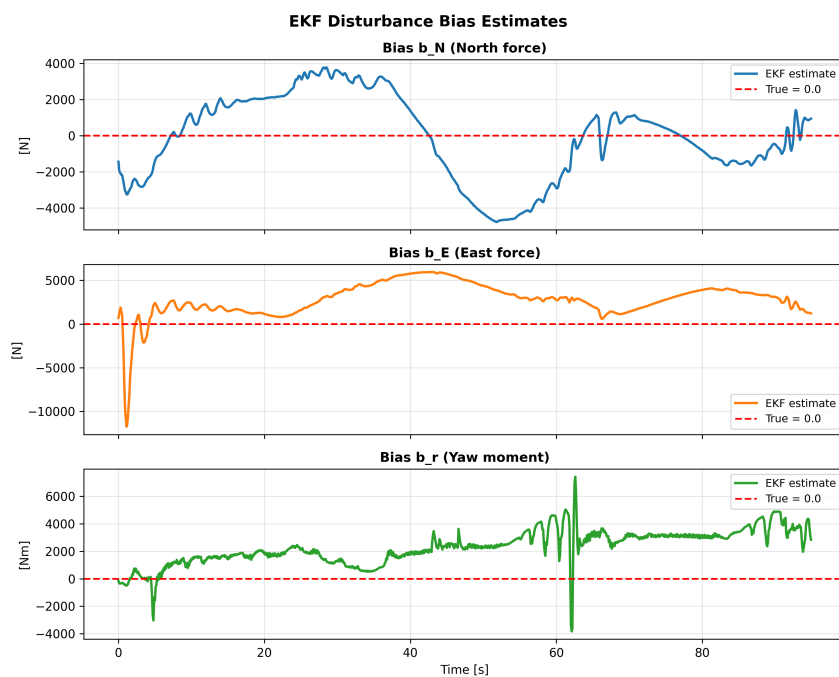


Figure 4.32: EKF bias estimate for constant disturbance.

4.5 The entire system real-world test on open water



Figure 4.33: Targa 46 boat used in the open water validation.

As a final evaluation of the system, some Real-World tests were conducted on a Targa 46 boat, see Figure 4.33. The vessel is a 13.93 m long and 2.95 m wide monohull vessel equipped with two Volvo Penta D8-IPS800s engines. The weight of the vessel was around 11 000 kg. Firstly, the calibration procedure was made to estimate the model parameters. Later, some simpler trajectories were tested in a controlled environment.

The tests were performed on two different occasions. After validating the system on the rig, preparations were made on the vessel, including reprogramming ECUs. The first validation occasion had problems with the rudder angle reading, leading to a faulty calibration and much debugging and problem solving at sea instead of logging. The second time seemed to go well with the calibration, but when controlling the vessel unpredictable behavior was present, leading to excessive problem solving at sea that took most of the assigned time.

4.5.1 Calibration first data collection

The following results seen in Figure 4.34, Figure 4.35, Figure 4.36, Figure 4.37 and Figure 4.38, were produced on a real marine vessel out at sea. The weather conditions were calm and suitable for data logging. The resulting parameter values can be seen in Table 4.4 and Table 4.5.

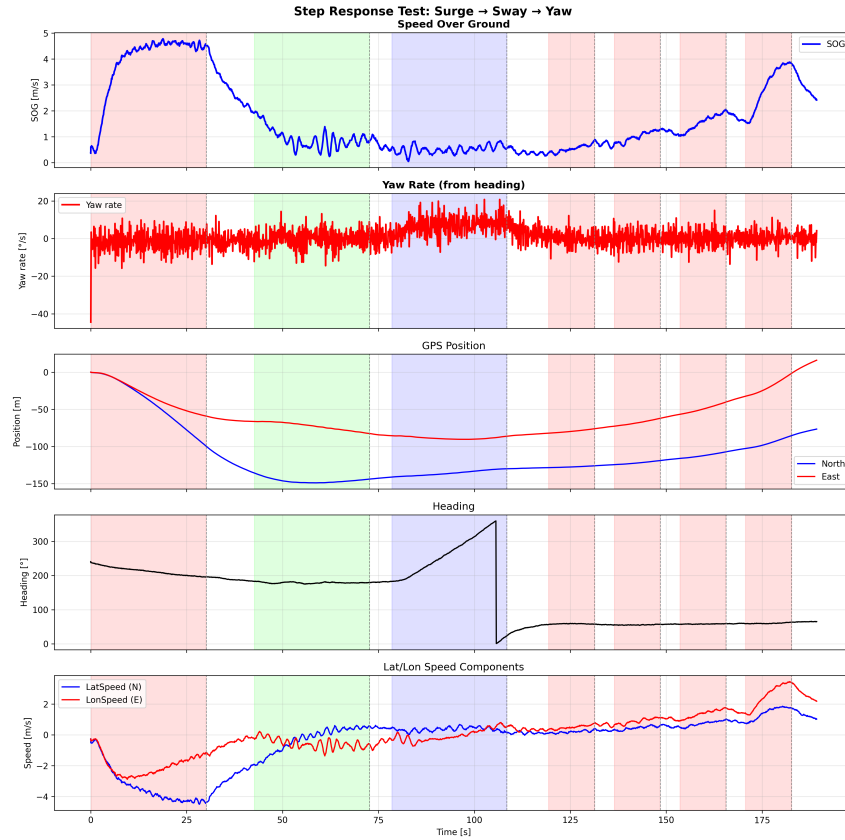


Figure 4.34: Real-World step-response including rpm-sweep.

Table 4.4: Parameters derived from step-response calibration (updated vessel test).

Category	Parameter	Value	Derivation
Velocity limits	$u_{\max}$	$\pm 4.58 \text{ m/s}$	Measured max surge SOG
	$v_{\max}$	$\pm 1.98 \text{ m/s}$	Measured max sway SOG
	$r_{\max}$	$\pm 20.79 \text{ deg/s}$	Measured max yaw rate
	$n_{\max}$	344.8 RPM	Steady-state surge RPM
Force limits	$X_{\max}$	2 487.6 N	$2 T_1^{\text{surge}}$ (592 RPM)
	$Y_{\max}$	1 925.5 N	$2 T_1^{\text{sway}} \sin \delta_{\max}$ (610 RPM)
	$N_{\max}$	2 387.7 Nm	$2 l_{y_t} k_1 \text{ RPM}_{\text{yaw}}$ (693 RPM)
Effective mass (acceleration)	m_{11}^{acc}	15 954.3 kg	$X_{\max}/a_0^u$, $a_0^u = 0.1559 \text{ m/s}^2$
	m_{22}^{acc}	15 782.8 kg	$Y_{\max}/a_0^v$, $a_0^v = 0.1220 \text{ m/s}^2$
	m_{33}^{acc}	250 238.0 kg m ²	$N_{\max}/\dot{r}_0$, $\dot{r}_0 = 0.009542 \text{ rad/s}^2$
Effective mass (coast-down)	m_{11}^{cst}	14 805.4 kg	$X_{\max}/ d^u $, $ d^u = 0.1680 \text{ m/s}^2$
	m_{22}^{cst}	12 922.8 kg	$Y_{\max}/ d^v $, $ d^v = 0.1490 \text{ m/s}^2$
	m_{33}^{cst}	48 006.9 kg m ²	$N_{\max}/ d^r $, $ d^r = 0.049737 \text{ rad/s}^2$
Combined mass (clamped)	m_{11}	15 379.8 kg	$0.5 m_{11}^{\text{acc}} + 0.5 m_{11}^{\text{cst}}$
	m_{22}	14 352.8 kg	$0.5 m_{22}^{\text{acc}} + 0.5 m_{22}^{\text{cst}}$
	m_{33}	149 122.5 kg m ²	$0.5 m_{33}^{\text{acc}} + 0.5 m_{33}^{\text{cst}}$

4. Results

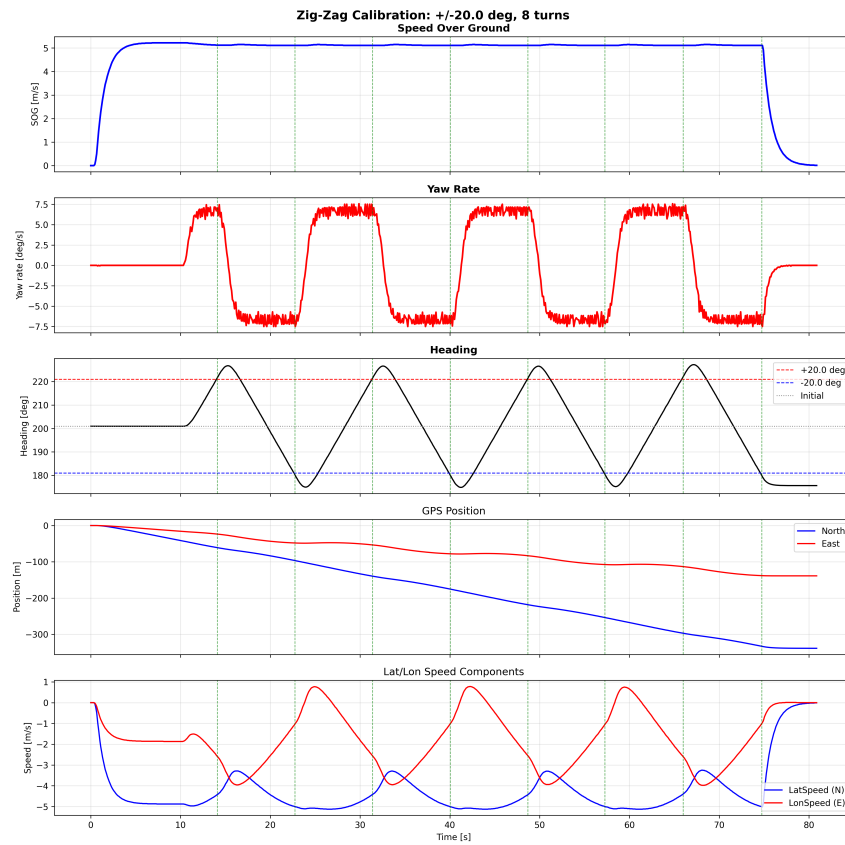


Figure 4.35: Zigzag calibration from rel life tests.

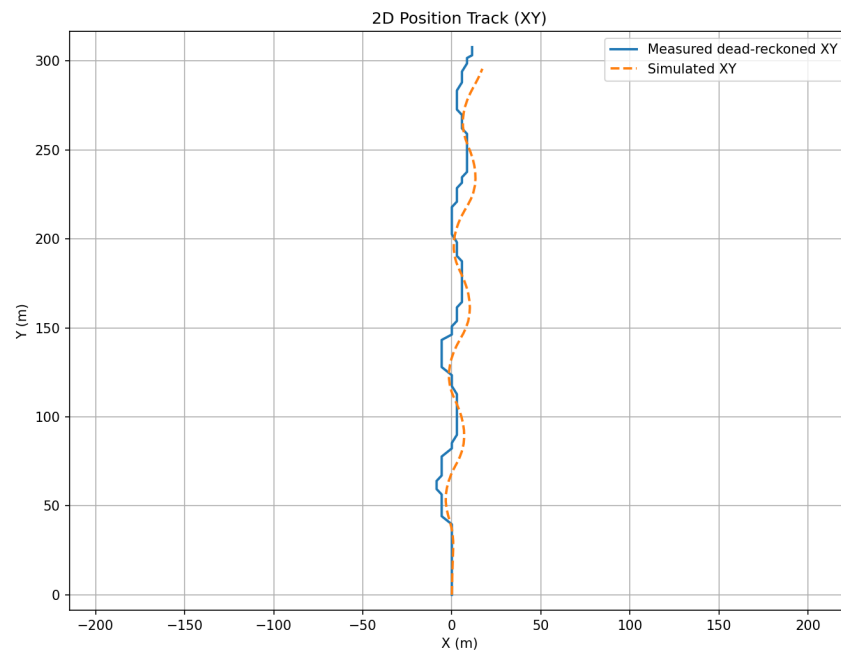


Figure 4.36: Simulation vs measured position.

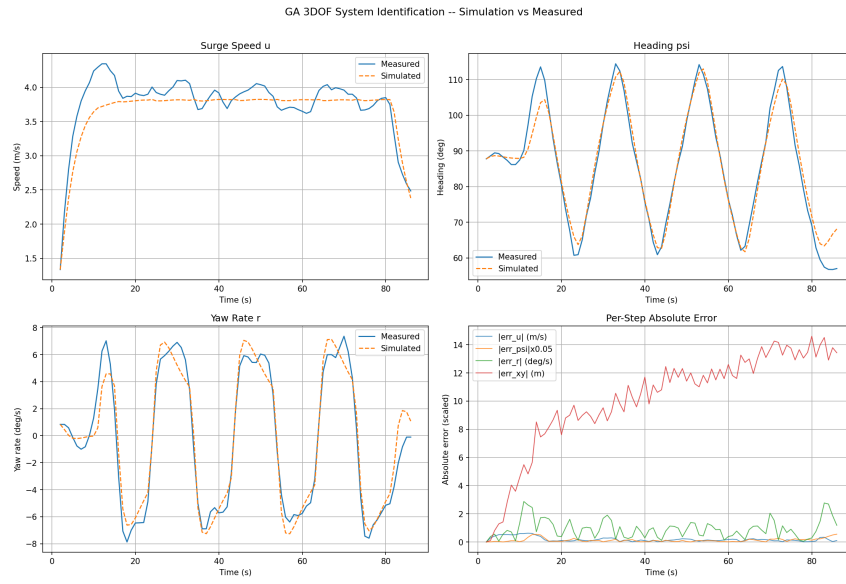


Figure 4.37: Real vs identified states.

Table 4.5: Identified Abkowitz maneuvering model parameters (GA fit with pinned calibration parameters).

Symbol	Description	Value
<i>Surge hydrodynamic derivatives</i>		
X_u	Linear surge damping	-1 217.47
X_{uu}	Quadratic surge damping	-311.93
<i>Sway hydrodynamic derivatives</i>		
Y_v	Linear sway damping	-7 530.11
Y_r	Yaw-induced sway force	42 925.96
<i>Yaw hydrodynamic derivatives</i>		
N_r	Linear yaw damping	-25 128.41
N_{rr}	Quadratic yaw damping	-44 435.43
N_v	Sway-induced yaw moment	-6 322.93
<i>Actuation-related parameters</i>		
k_1	Thrust gain coefficient (pinned)	3.4716
k_2	Nonlinear thrust coefficient	500.00
Y_d	Differential thrust sway gain	485.26
N_d	Differential thrust yaw gain	336.56
l_{yt}	Lateral thruster lever arm (pinned)	0.50 m

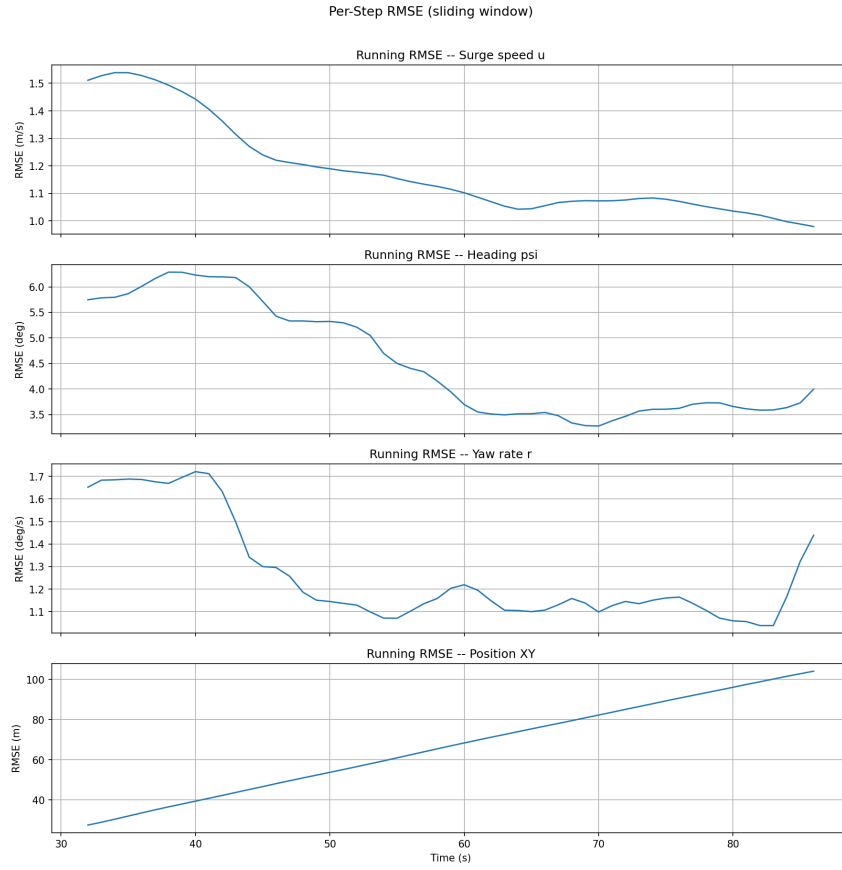


Figure 4.38: RMSE for position and heading between the simulated and measured trajectory.

These tests demonstrate the potential of the calibration procedure for estimating model parameters using a data-driven, constrained genetic algorithm. The results are detailed in the following subsections.

4.5.2 Calibration second data collection

This calibration was performed following the results presented in subsection 4.5.1, under calmer environmental conditions to reduce disturbances and improve parameter consistency. The step-response experiment used for parameter estimation is shown in Figure 4.39, with the resulting calibrated parameters summarized in Table 4.6 and Table 4.7. Model validation is shown in Figure 4.41 and Figure 4.42, with RMSE presented in Figure 4.43.

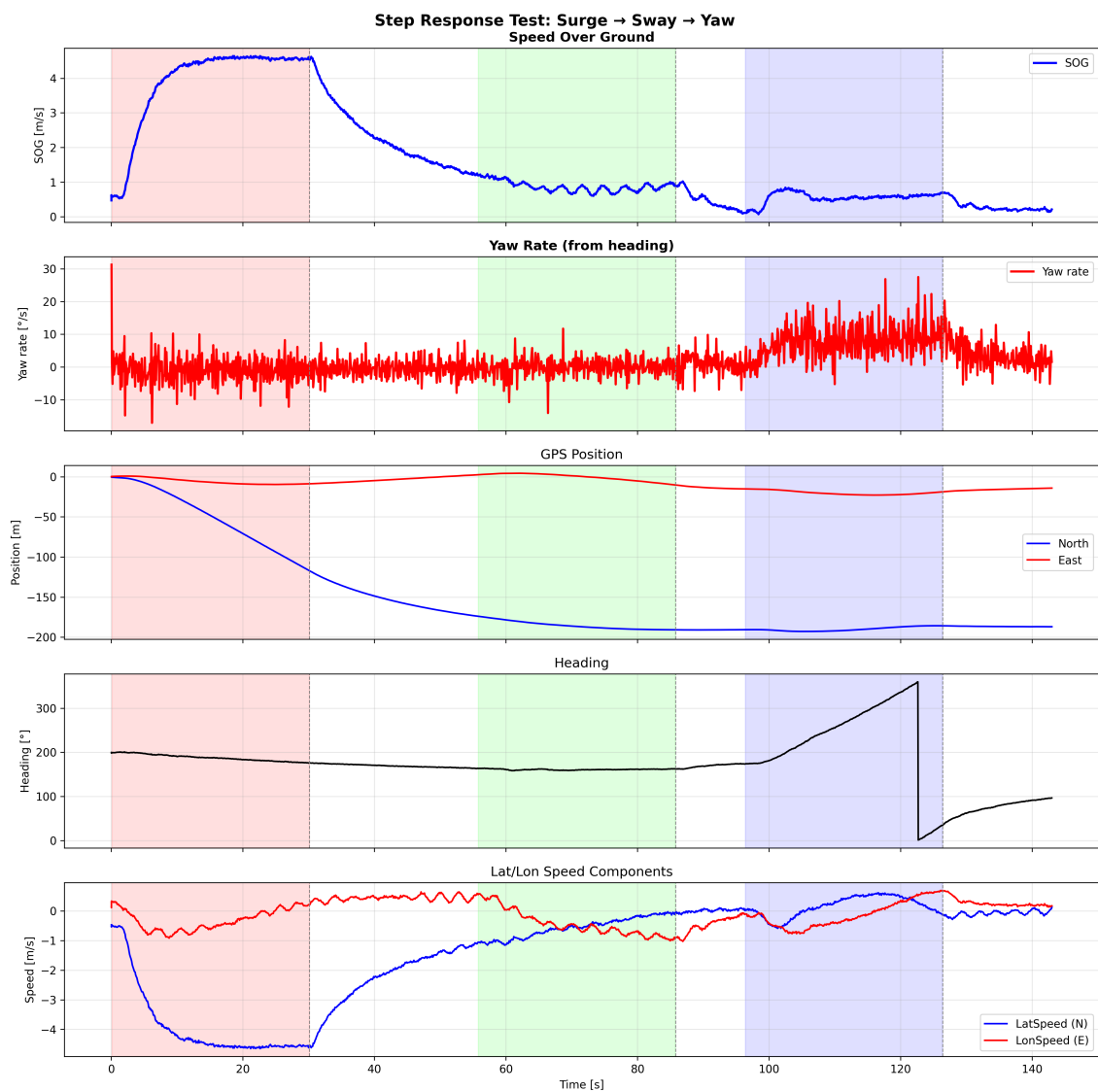


Figure 4.39: Step-response calibration from real vessel.

Table 4.6: Parameters derived from step-response calibration (updated vessel test).

Category	Parameter	Value	Derivation
Velocity limits	$u_{\max}$	$\pm 4.64 \text{ m/s}$	Measured max surge SOG
	$v_{\max}$	$\pm 1.23 \text{ m/s}$	Measured max sway SOG
	$r_{\max}$	$\pm 27.47 \text{ deg/s}$	Measured max yaw rate
	$n_{\max}$	594.9 RPM	Steady-state surge RPM
Force limits	$X_{\max}$	4 612.1 N	$2 T_1^{\text{surge}}$ (595 RPM)
	$Y_{\max}$	2 000.4 N	$2 T_1^{\text{sway}} \sin \delta_{\max}$ (610 RPM)
	$N_{\max}$	2 522.5 Nm	$2 l_{y_t} k_1 \text{ RPM}_{\text{yaw}}$ (693 RPM)
Effective mass (acceleration)	m_{11}^{acc}	12 875.8 kg	$X_{\max}/a_0^u$, $a_0^u = 0.3582 \text{ m/s}^2$
	m_{22}^{acc}	68 273.0 kg	$Y_{\max}/a_0^v$, $a_0^v = 0.0293 \text{ m/s}^2$
	m_{33}^{acc}	88 342.7 kg m ²	$N_{\max}/\dot{r}_0$, $\dot{r}_0 = 0.028554 \text{ rad/s}^2$
Effective mass (coast-down)	m_{11}^{cst}	11 579.5 kg	$X_{\max}/ d^u $, $ d^u = 0.3983 \text{ m/s}^2$
	m_{22}^{cst}	12 227.4 kg	$Y_{\max}/ d^v $, $ d^v = 0.1636 \text{ m/s}^2$
	m_{33}^{cst}	67 917.6 kg m ²	$N_{\max}/ d^r $, $ d^r = 0.037141 \text{ rad/s}^2$
Combined mass (clamped)	m_{11}	12 227.6 kg	$0.5 m_{11}^{\text{acc}} + 0.5 m_{11}^{\text{cst}}$
	m_{22}	40 250.2 kg	$0.5 m_{22}^{\text{acc}} + 0.5 m_{22}^{\text{cst}}$
	m_{33}	78 130.1 kg m ²	$0.5 m_{33}^{\text{acc}} + 0.5 m_{33}^{\text{cst}}$

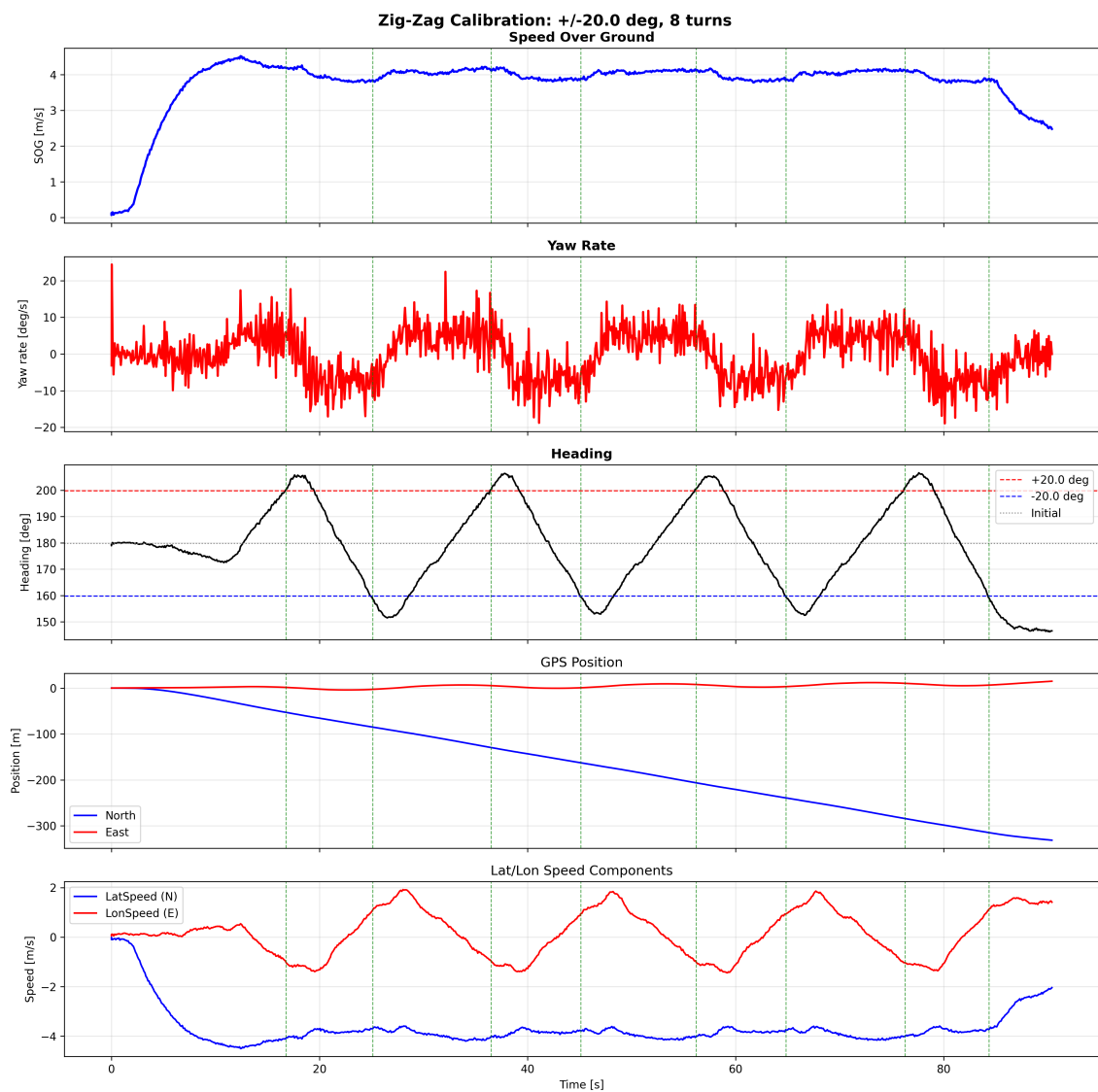
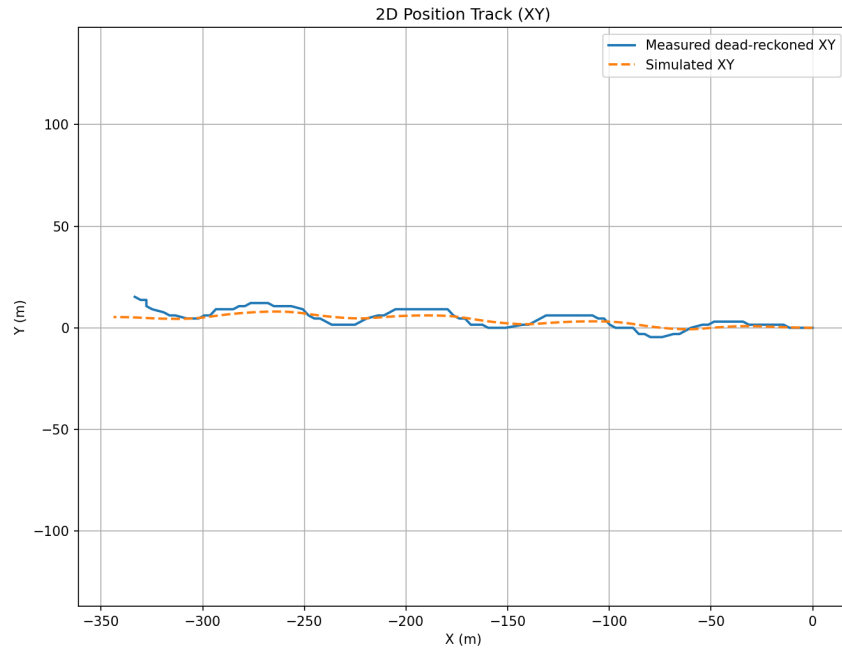


Figure 4.40: Zigzag calibration 3.2.

Table 4.7: Identified Abkowitz maneuvering model parameters (GA with equilibrium-consistent bounds).

Symbol	Description	Value
<i>Surge hydrodynamic derivatives</i>		
X_u	Linear surge damping	-2 003.50
X_{uu}	Quadratic surge damping	-383.07
<i>Sway hydrodynamic derivatives</i>		
Y_v	Linear sway damping	-1 403.42
Y_r	Yaw-induced sway force	-4 614.51
<i>Yaw hydrodynamic derivatives</i>		
N_r	Linear yaw damping	-20 419.26
N_{rr}	Quadratic yaw damping	-42 200.33
N_v	Sway-induced yaw moment	-20.93
<i>Actuation-related parameters</i>		
k_1	Thrust gain coefficient (pinned)	3.5903
k_2	Nonlinear thrust coefficient	10.80
Y_d	Differential thrust sway gain	2 256.32
N_d	Differential thrust yaw gain	27.74
l_{yt}	Lateral thruster lever arm (pinned)	0.50 m

**Figure 4.41:** Real measured position track vs simulated position track.

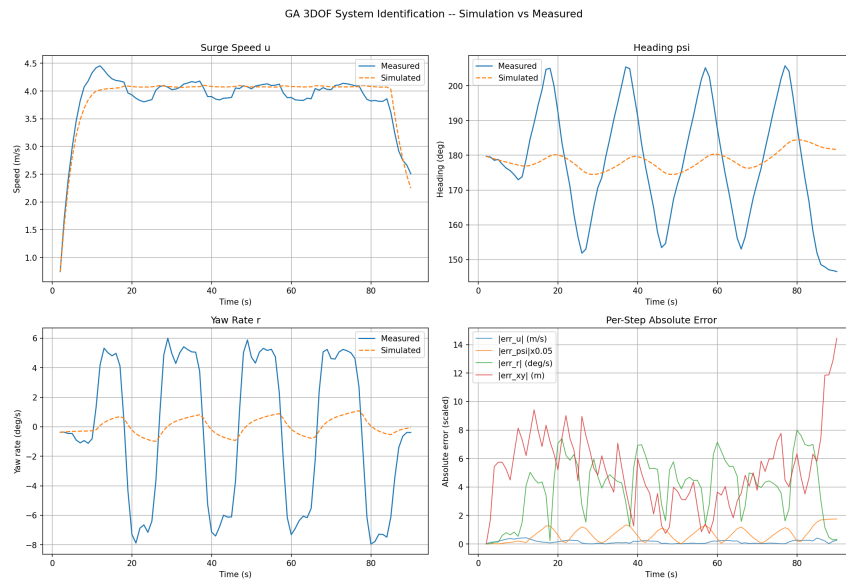


Figure 4.42: Measured vs simulated states from GA identification.

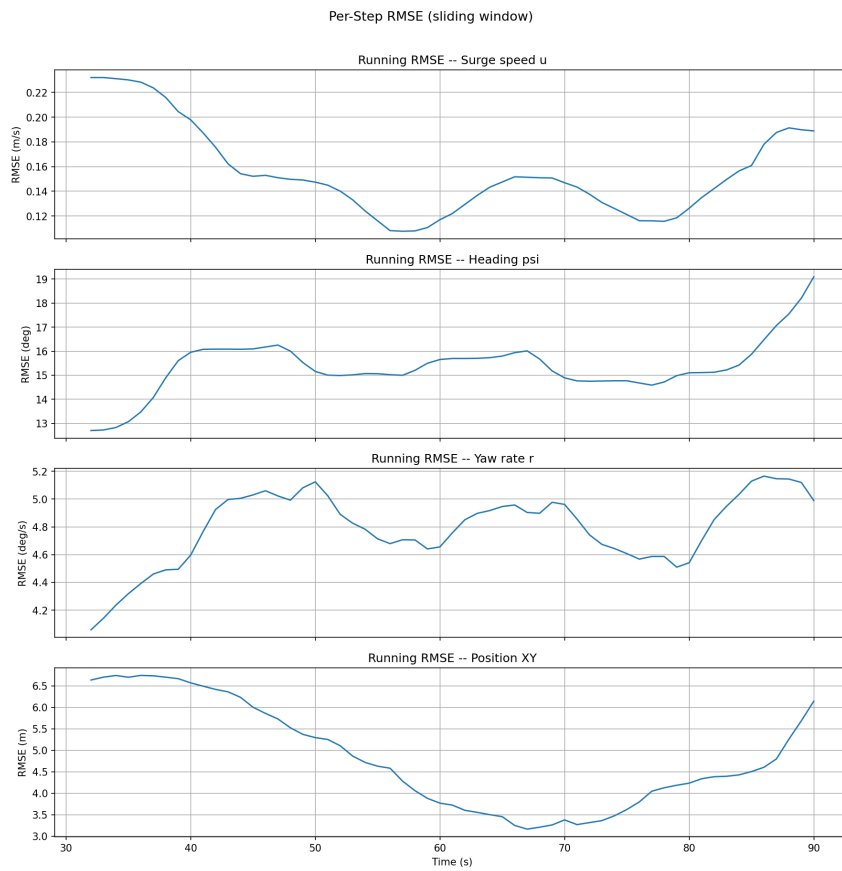


Figure 4.43: RMSE for position and heading between the simulated and measured trajectory.

4.5.3 Real model predictive controller tests

As a final evaluation using the simulation rig, the tuned model was tested in combination with the MPC and EKF. The performance achieved was limited, and due to time constraints as well as discrepancies between the real system and the simulation rig, this represents the final level of performance obtained within the scope of this project. The system demonstrates potential, but requires further refinement. However, its performance is far from optimal in the open-water test, as shown in Figure 4.44.

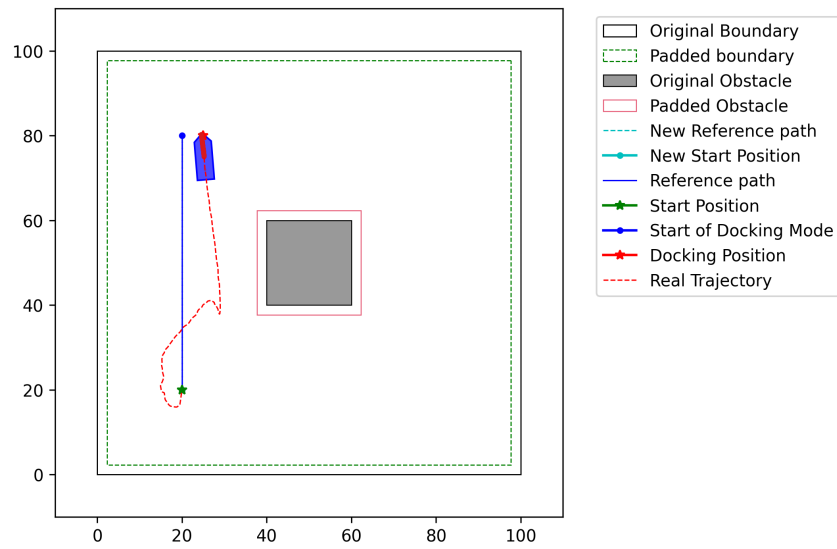


Figure 4.44: Trajectory obtained during the full system validation.

The solver time results are promising, even though the system has not been fully optimized, indicating that real-time feasibility is achievable (Figure 4.45).

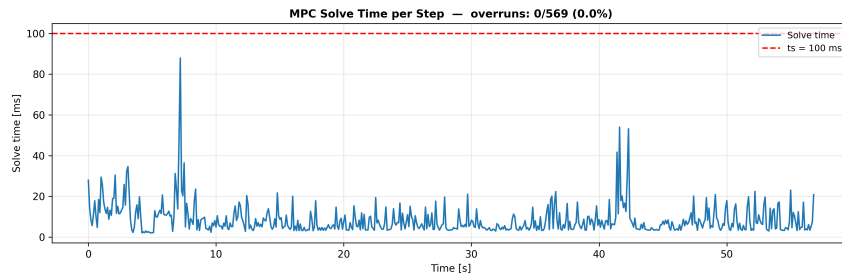


Figure 4.45: Solver time during the real system test.

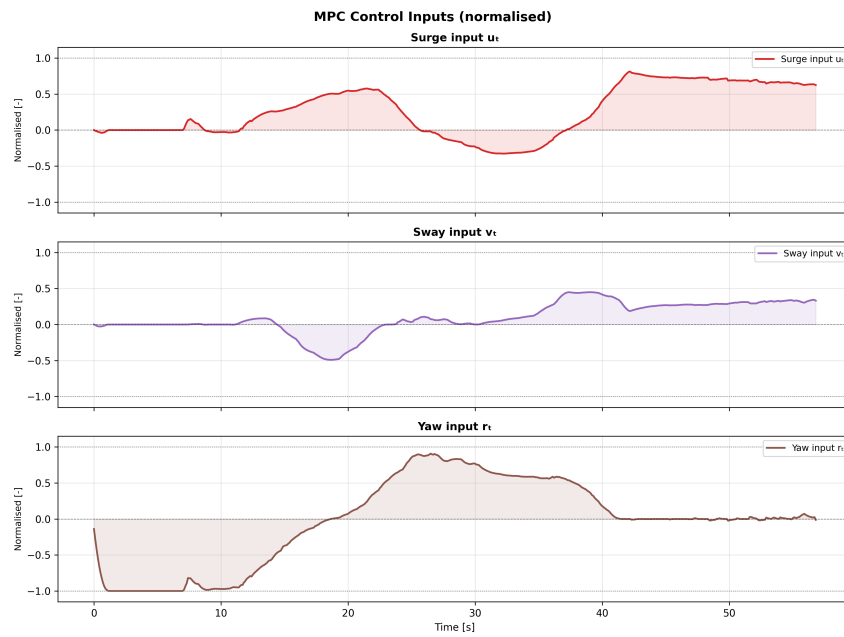


Figure 4.46: Computed control inputs during the test.

The computed control inputs shown in Figure 4.46 are smooth and exhibit the expected coupled behavior between actuators. Furthermore, the heading error (Figure 4.48) and cross-track error (Figure 4.47) show consistent and predictable patterns that align with the observed trajectory deviations.

This suggests that the primary limitation is not instability or numerical issues, but rather suboptimal MPC tuning and potential model mismatch affecting tracking performance.

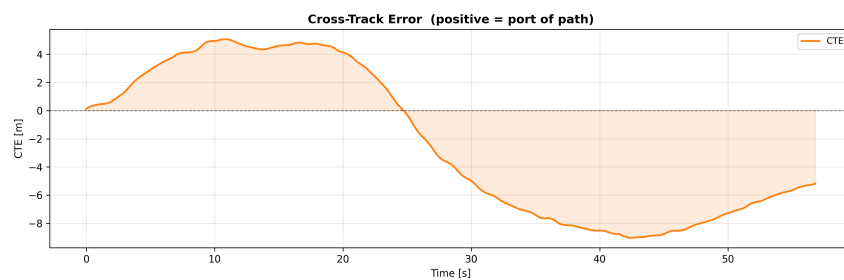


Figure 4.47: Cross-track error (CTE) during the experiment.

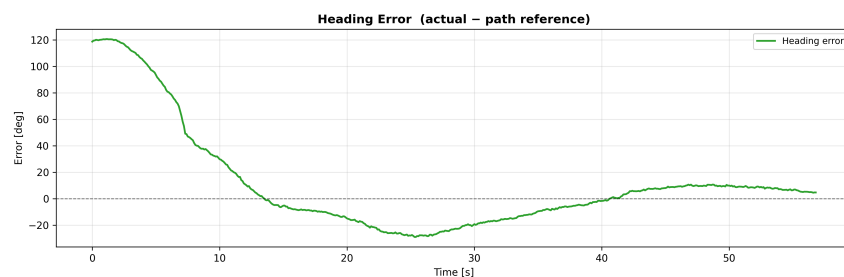


Figure 4.48: Heading error during the experiment.

4.6 Control allocation

This section presents the results obtained from the control allocation algorithm. The performance of the allocator is evaluated based on its ability to distribute the provided thrust commands among the available thrusters while satisfying the physical and operational constraints.

4.6.1 Twin thrusters at the stern in simulation rig

With thruster positioning at $l_x = 4.0$ m and $l_y = 0.8$ m, the resulting trajectory is shown in Figure 4.49. The vessel follows the reference path but exhibits minor overshoot at the corners and during the final docking phase. Figure 4.50 displays the MPC-requested forces in surge, sway, and yaw, alongside the achieved force output from the control allocator. Figure 4.51 shows the RPM and steering angle commands for both port and starboard thrusters.

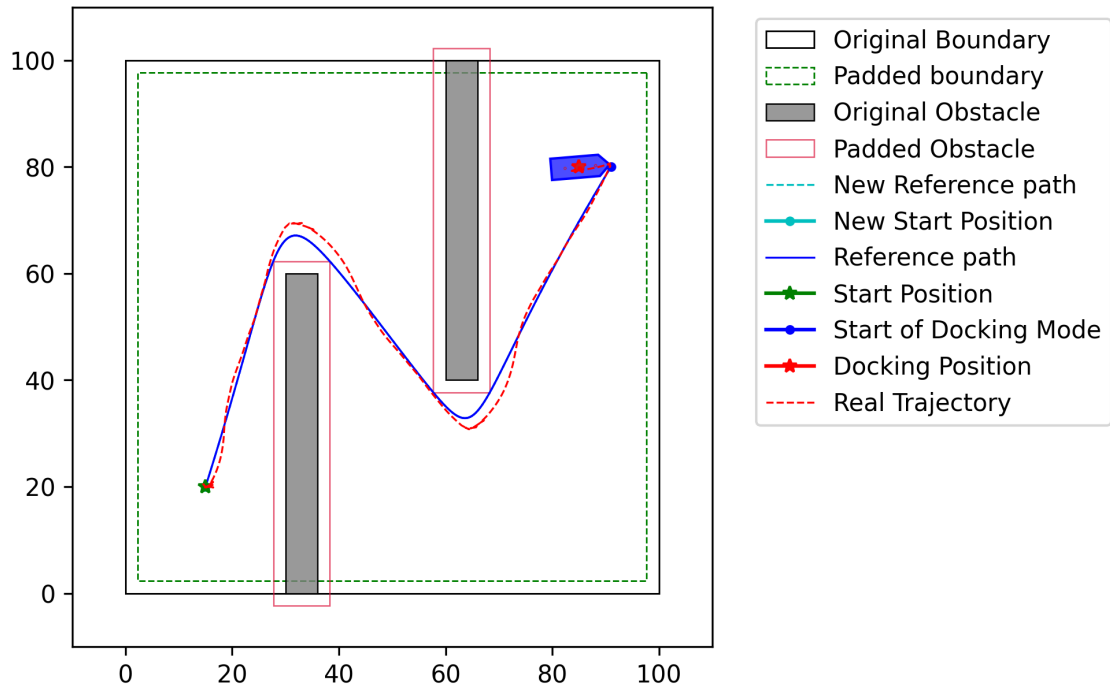


Figure 4.49: The resulting path of the allocator outputs to the marine vessel.

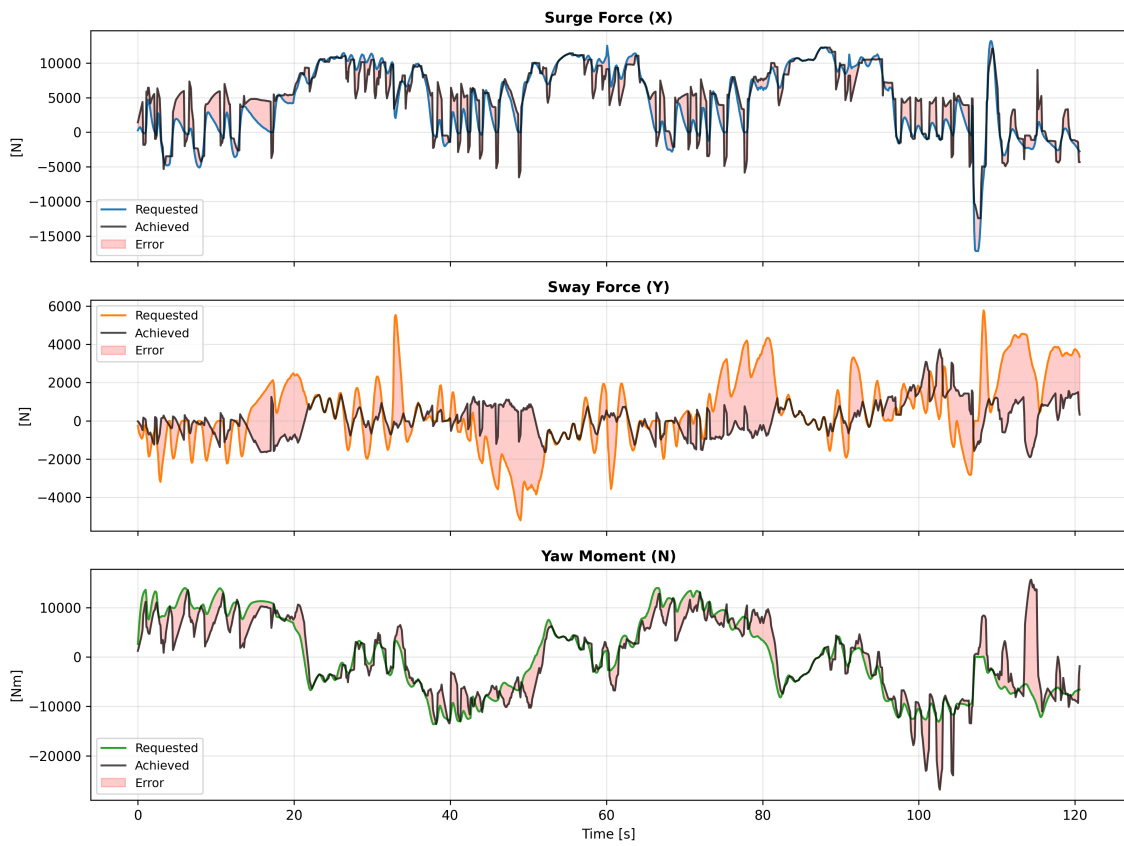


Figure 4.50: Result of the allocator forces provided after request from MPC with same force limits.

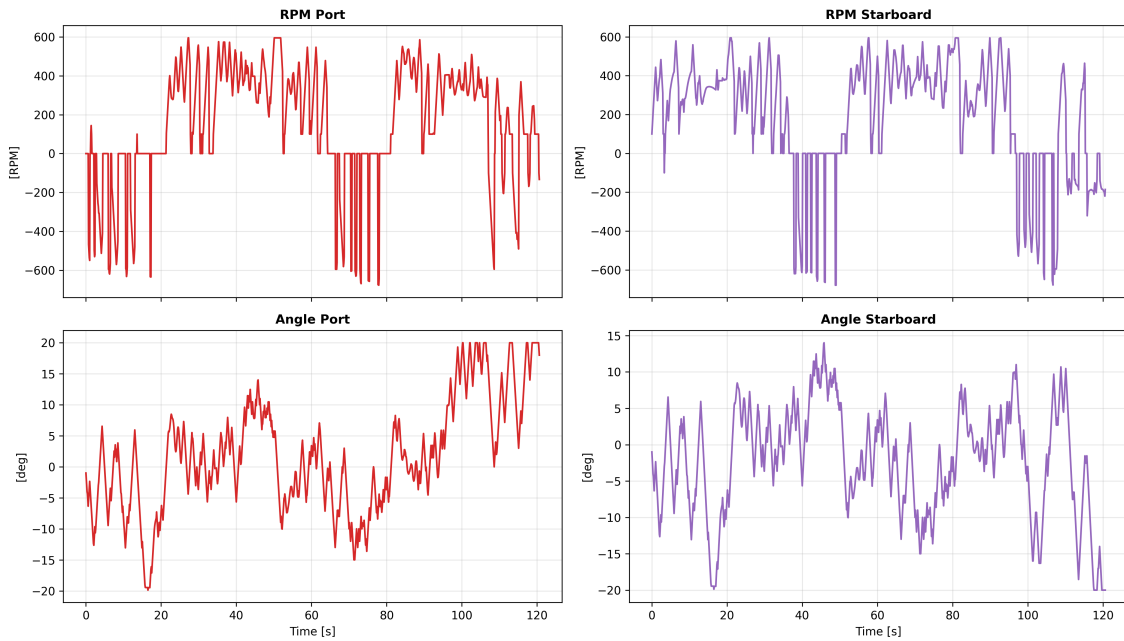


Figure 4.51: Result of the allocator commands with angle δ and RPM of both thrusters.

5

Discussion

The following chapter aims to reflect on and analyze the results and different sections of this thesis. The discussion considers the limitations, progress, results, and the overall scope of the project, with the objective of providing a comprehensive overview of autonomous docking and its potential role in future vessel maneuvering.

5.1 Dynamic model

The selected dynamic model, MAM, is a simplified representation of the Abkowitz model that performs well for the specific case studied in this thesis. The main motivation for choosing this model was to limit the number of hydrodynamic parameters while maintaining adaptability to multiple vessels. With these limitations, uncertainty in accuracy and performance is introduced, which needs to be taken into consideration to maintain robust performance.

To mitigate these limitations, the model parameters were identified using real-world data from the specific vessel in combination with an optimization algorithm such as GA. This approach reduces the risk of poor model performance, although it does not completely eliminate it. As demonstrated in the results, this method provides good performance and successfully adapts towards its surroundings. By integrating real measurements, the parameters can be closely tuned to match the behavior of the actual vessel dynamics, aligning with a primary objective of this thesis.

Comparison between the Fossen model and MAM showed that the choice of dynamic model had a notable impact on the simulated vessel behavior. Consequently, MAM was selected as the final representation.

The optimization method selected for parameter identification in this thesis was GA. This optimizer is well suited for this kind of problem since it has a global search span. Other optimization methods, such as PSO, were considered in the early stages of the project. However, when comparing the different optimization methods, the result showed that the choice of optimizer did not have a critical effect on the performance compared to the choice of dynamic model. When applying the same model with GA and PSO, performance was comparable and the primary difference was the resulting parameters, not the overall accuracy.

5.2 Model predictive controller

The MPC is an appropriate controller for this type of application. Its capability to predict and plan for obstacle avoidance is crucial in docking scenarios, particularly in busy harbours. The MPC's estimates appear to be accurate and converge towards the reference trajectory. Model simplifications and accumulated assumptions

contribute to uncertainties, and the system struggles in more challenging situations. In cases involving small, sharp turns, the system tends to slow down significantly or even stall.

The model dependency of the MPC is, understandably, one of the most challenging aspects of this controller. The overarching aim of this thesis is to develop a controller that is easy to implement and based on an identified model. However, the absence of accurate hydrodynamic data introduces additional uncertainties that are difficult to quantify. As a result, the system will be evaluated primarily on its overall performance rather than on the performance of each individual component.

The MPC shows good path following and control of the system, as seen in the result. A notable vulnerability in the MPC is its sensitivity to minor parameter variations. Suboptimal weight values significantly degrade vessel behavior. Consequently, weight selection requires careful tuning.

The performance of the cost function is influenced by the chosen weighting parameters. While the current implementation works, it requires a more structured design to achieve more robust vessel behavior. Improving this part of the project is essential for reliable operations in real environmental conditions with external disturbances and surrounding influences.

The solver subsection 2.4.7 appears to be sufficiently fast for real-time execution of the optimization problem presented in Figure 4.45. This capability is crucial for deploying an online system of this scale. While the complexity of the optimization procedure could be substantially reduced to accommodate simpler embedded controllers, this lies beyond the scope of the present thesis. For debugging purposes and for the initial design of the controller, it would be advantageous to employ a more straightforward solver with a shorter learning curve, thereby mitigating initial confusion and the risk of being overwhelmed by complexity and long build times.

5.3 Extended Kalman Filter

The EKF demonstrated robust performance in controlled simulation environments. However, since the EKF relies on the identified vessel model, its estimation accuracy is inherently dependent on the quality of the model. Any mismatch between the model and the true vessel dynamics, including unmodelled effects, may propagate into the state estimates and degrade performance.

To account for such discrepancies, bias terms were incorporated into the EKF formulation. These terms represent the accumulated effect of disturbances acting on the vessel, including environmental forces and model uncertainties. While this approach improves robustness against modeling errors, it introduces additional tuning complexity. In particular, the selection of appropriate bias dynamics and associated noise covariance matrices is challenging due to the absence of ground-truth disturbance measurements.

Overall the EKF appears to be a viable state observer for the system. However the added complexity complicates system tuning and makes it more difficult to decouple and validate individual subsystems, especially in real-world experiments. Independent validation is ideal, the EKF's reliance on the identified vessel model makes fully decoupled validation infeasible.

Finally, due to time constraints, the EKF was not a primary focus during the real-life

testing, where priority was instead given to system-level validation.

5.4 Control allocation

The control allocation algorithm demonstrates good performance with its limitations, both physical and operational. However, achieving a more reliable and optimal control allocation would require additional specific information about the vessel, such as the exact location of the CoG and the position of the thrusters relative to the CoG.

With access to more detailed vessel-specific information, the control allocation algorithm could provide even more optimal force distribution and decision making. With the current limitation of such information, the control allocator is capable of distributing the force commands from the MPC towards thruster RPM and angle. However, due to uncertainty in the thruster location relative to the vessel's CoG, the generated commands are not fully optimal, which affects the accuracy and performance of the vessel's maneuvering.

Further performance improvements could be achieved by tighter coupling of the control allocator and MPC framework. This integration would allow the controller to have access to the system limitations and improve the feasibility and control accuracy. Additional performance improvements could be achieved through a more detailed understanding of the thruster force characteristics, such as propeller curves and the thrust-RPM relationship.

5.5 Whole system

The complete system can be separated into two distinct validation stages, as their results differed considerably.

5.5.1 Simulation

The simulations produced encouraging results, where a dynamic model could be identified and subsequently controlled using the full pipeline. The system succeeded in deriving a reasonable data-driven model from only simple initial conditions, combined with straightforward MPC and EKF tuning. The scenarios evaluated in section 4.4 were deliberately designed to reflect typical low-speed docking maneuvers. For the MPC, the plant was replaced by a separate, imperfect model in a feedback loop, effectively treated as a black box. All new measurements and inputs were processed by the simulation rig's own dummy CAN components, providing a fairly realistic in-house test environment. Setting this up correctly, and making the MPC and system identification operate reliably together with the Multilink communication, required substantial effort. Overall, this represented a step closer to real-world trials. However, the dummy validations initially exhibited issues that were not fully addressed, such as a separate black box for the control inputs, which made the control actions more difficult to track. Additional mistakes in the construction of the simulation environment caused confusion and forced revisions of the original system concept. Model mismatches posed a major challenge that the EKF was intended to mitigate, but as system complexity increased, it became difficult to keep the different subsystems decoupled.

The actual mathematical model of the simulation rig was not available in the documentation, meaning no true physical parameters of the rig were known. Instead, the setup relied solely on observed simulation performance, resulting in extensive tuning without a proper model to serve as a “true baseline.” A predefined, high-fidelity vessel model with dynamics close to real-world behavior would have been advantageous for the system identification stage, as demonstrated for instance in [26].

The auto-docking system was thoroughly tested on this simulation rig and tuned based on the simulator’s feedback. This however, introduced cumulative simplifications that remove many of the real-world dynamics and behaviors present in actual systems.

5.5.2 Real tests

The real-world experiments provided an indication of the method’s potential. The similarity between the step responses obtained from the test rig Figure 3.1 and the real-life calibration results Figure 4.34 and Figure 4.39 suggests that this approach could be a suitable step toward a gray-box identification, where limits are determined based on actual measurements. The calibration procedure itself behaved in a controlled and predictable manner, although the duration of the procedure could be shortened, which is left as a future improvement. Comparing the identified parameters in Table 4.1 with those in Table 4.4 shows that the velocity limits are quite similar, whereas the same does not hold for the force limits. At comparable rpm values, the maximum force in the real tests is only 14–25% of the forces estimated from the rig. This discrepancy arises from poor thrust parameter estimates, which rely on the k_1 and k_2 parameters. These, in turn, are highly sensitive to accurate assumptions about the propeller–thrust relationship as well as the estimated masses. The yaw calibration data appear to be rather noisy, which may lead to poor model estimates. This could be alleviated by applying more appropriate filtering to the yaw rate. Another potential source of mismatch is the lack of sufficient excitation in sway, which can result in inaccurate sway-related parameters, as indicated in Figure 4.34 and Figure 4.39. The requirement for calm water conditions and the reliance on stabilizing to zero velocity represent weak points in the calibration procedure. Adjusting the mass parameters based on the measured acceleration and subsequent coasting behavior could improve the accuracy of the effective mass estimates. The MPC tuning was not completed within the limited sea trial timeframe. This likely contributed to the performance gap between simulation and real-world results.

5.6 Future work

With the result and performance of this thesis, the combination of SI, MPC, and control allocation shows strong potential to improve autonomous docking and maneuvering of the vessel. The methodologies and findings presented can be applied to a wide range of marine vessels, where simplified yet accurate dynamic models are required for autonomous applications.

One important subject for future work is the development of a more general and robust MPC framework, including self-calibration of adaptive weighting matrices. Future work should investigate optimization algorithms for tuning MPC weights. This was explored to a limited extent with some success, though not sufficiently

to incorporate into the final result. A successful implementation of adaptive MPC weights would significantly improve system generality and enable application to a broader class of vessels.

The system identification step could be extended by introducing an online calibration mechanism to continuously refine the parameters of the dynamic vessel model. Similar to the disturbance estimation performed by the EKF, such an approach would allow the model to adapt to changing environmental and operational conditions. This is particularly important in marine environments, where weather and operating conditions can change rapidly. Therefore, online adaptation of the model parameters could further improve both modeling accuracy and control performance of the vessel.

Furthermore, integration of external object identification such as LiDAR or radar would enable real-time object detection and obstacle avoidance. This would allow the vessel to see real obstacles and navigate around them. With the current system, obstacles are assumed to be already known. Extending the current system with this would therefore significantly improve the maneuvering safety and more robustness, particularly in harbour environment where often there are multiple dynamic obstacles but also static obstacles, such as other vessels, piers, shallow water, shorelines, and submerged structures. Such integration is essential and a critical step towards a fully autonomous and safe docking operation.

When implementing the proposed system on a real marine vessel and conducting field experiments at sea, several limitations were observed in both the MPC strategy and the control allocation scheme.

The MPC path following algorithm exhibited sensitivity when the vessel deviated from the reference trajectory. In particular, when the vessel was slightly off the desired path or had already passed the goal point, the control tended to generate excessive control forces in an attempt to compensate for the position error. This often resulted in aggressive corrective action and overshoot around the target position. These observations indicate that the current cost function and weighting strategy in the MPC formulation are not sufficiently robust to handle rougher conditions. Further work should therefore focus on improving the tuning of the MPC weights and a more robust path-following strategy. For example, as mentioned in section 5.2 adaptive weights tuning, switch weighting strategies based on the vessel's position to handle suboptimal scenarios.

When applying the control allocator, the actuator forces requested by the MPC were limited during real-world tests. While this improved safety and feasibility, an important issue was identified in the allocation strategy. In some cases, the allocator produced opposing thrust commands - positive surge thrust in one thruster and negative thrust in the other - to achieve the desired combination of forward motion and rotational torque. Although this is a mathematically optimal solution with respect to the requested generalized forces, such behavior reduces efficiency from both energy and mechanical perspectives.

Ideally, when requesting forward motion with rotation, the system should prioritize adjustments of thruster angles rather than differential thrust involving reversal of direction. This issue arises from the interaction between the force demands generated from the MPC and the optimization criteria used in the control allocator. The

same behavior was observed in both Python-based simulations and in the testing rig, indicating that it is more of a structural issue in the control design rather than a hardware-specific issue. To address this, the control allocation strategy was revised. The updated implementation introduces stronger penalties on reverse thrust, enforces directional constraints when forward motion is requested, and includes gear-change penalties and delays to discourage rapid sign changes in RPM. Further work should therefore investigate additional improvements to the control allocation strategy to better reflect practical operational constraints and efficiency considerations. This could be done by refining penalty formulations, further prioritizing thrust vectoring over magnitude changes, and tightening the integration between the MPC and the control allocator, as discussed in section 5.4.

Overall, addressing these challenges will be essential for a more robust, efficient, and reliable autonomous docking performance in real marine environments.

6

Conclusion

This thesis has investigated the challenges in autonomous docking of marine vessels and provided insight into this research area. The focus has been on integrating a general vessel model capable of representing real-world behavior, with the goal of accurate path following and docking maneuvering. A key objective was to identify a suitable dynamic model that could reproduce real vessel behavior and support autonomous maneuvering. To achieve this in an efficient and accurate way, a combination with real-world data, dynamic modeling, and optimization techniques was employed.

The dynamic model used in this thesis uses the necessary parameters to capture vessel behavior that is observed in real-world logged data. With appropriate parameter identification, the model shows the ability to replicate real vessel dynamics with reasonable accuracy, despite being a simplified representation.

The results show that the chosen dynamic model captures the vessel motion accurately for control purposes. When integrating within MPC framework, the designed cost functions are able to successfully navigate the vessel around obstacles and guide it safely towards the final docking positions. These results demonstrate the effectiveness of combining system identification, MPC, and control allocation for autonomous docking applications.

The performance of the system can be further improved through more advanced integration within the MPC framework. Optimizing the weighting matrices of the MPC cost function using automated optimization algorithm could increase the performance of tracking and controlling the vessel. Future work should therefore consider this sort of implementation of optimizer algorithm for automatic weight tuning, potentially combining it with online calibration methods. Such approaches would potentially allow better performance and increased robustness under different operating and environmental conditions.

Experimental testing on a real marine vessel revealed additional practical challenges. More specifically, issues related to rudder angles response, indicating that the vessel had difficulty accurately assessing its turning capability. As a result, the system struggled to consistently follow the desired heading during path-following operations. However, during docking mode where the vessel operates with a more constrained and stable heading, the system demonstrated improved turning capability and more reliable motion towards the final position.

Overall, the real-world experiments highlight both the potential and the limitations of the proposed approach. While the results demonstrate that this type of system and maneuvering strategy is feasible for autonomous docking, further improvements are required. In particular, better smoothing and tuning of the MPC weights are

necessary to ensure stable and smooth behavior, especially when the vessel deviates from the optimal path.

The research questions formulated in this thesis can be addressed as follows. A generalized dynamic model for vessels in the 20–100 ft range can be identified and calibrated using simple decoupled maneuvers such as step-response and zigzag tests. Further computations using Genetic Algorithm enables refinement of the hydrodynamic parameters, although the approach remains dependent on the quality and excitation of the available data. The required measurement data was shown to consist of standard onboard signals, including GPS-based position and velocity together with actuator commands, which can be processed and filtered in real time to support both system identification and control. An MPC-based control architecture, supported by an EKF serving as a state observer and disturbance compensator demonstrated reliable trajectory tracking and convergence to docking states in simulation. Additional development time and enhanced robustness are required before it can be reliably used in real-world conditions. A separate control allocation module, operating independently of the main control loop, appears to be a suitable approach, but it was not sufficiently validated in real-world tests.

This approach aligns well with industrial needs for modular and transferable control systems deployable across multiple vessel platforms. Despite these limitations, the presented framework demonstrates a promising step toward scalable and adaptable autonomous docking systems for leisure vessels, reducing reliance on vessel-specific tuning and enabling broader adoption of marine autonomy.

Bibliography

- [1] Y. Dai and K. Lee, “Deep reinforcement learning-based autonomous docking with multi-sensor perception in sim-to-real transfer,” *Processes*, vol. 13, no. 9, 2025, cited by: 0; All Open Access, Gold Open Access. [Online]. Available: <https://www.scopus.com/inward/record.uri?eid=2-s2.0-105017124561&doi=10.3390%2fpr13092842&partnerID=40&md5=43c395cd427283ea1f6db5a957195421>
- [2] E. Laitala and E. Strömdahl, “Autonomous docking: Trajectory planning and dynamic route adaptation for autonomously docking a marine vessel using model predictive control,” Master’s thesis, Chalmers University of Technology, 2024.
- [3] R. Kotian, P. Umesh, and K. Gangadharan, “Real-time implementation of pid control for unmanned surface vessel: A practical approach,” 2023, Conference paper, cited by: 5. [Online]. Available: <https://www.scopus.com/inward/record.uri?eid=2-s2.0-85172661004&doi=10.1109%2fCONECCT57959.2023.10234784&partnerID=40&md5=4ea66d765943d775dcdf68cbb1e54240>
- [4] S. J. Lexau, A. M. Lekkas, and M. Breivik, “Nonlinear pid control for automatic docking of a large container ship in confined waters under the influence of wind and currents**this work was supported by the research council of norway under project 309230 (sfi autoship).” *IFAC-PapersOnLine*, vol. 58, no. 20, pp. 265–272, 2024, 15th IFAC Conference on Control Applications in Marine Systems, Robotics and Vehicles CAMS 2024. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S2405896324018202>
- [5] A. B. Martinsen, A. M. Lekkas, and S. Gros, “Reinforcement learning-based nmpc for tracking control of asvs: Theory and experiments,” *Control Engineering Practice*, vol. 120, p. 105024, 2022. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0967066121002823>
- [6] F. Salem, M. I. Mossad, and M. Awadallah, “A comparative study of mpc and optimised pid control,” *International Journal of Industrial Electronics and Drives*, vol. 2, p. 242, 01 2015.
- [7] T. I. Fossen, *Handbook of Marine Craft Hydrodynamics and Motion Control*, 2nd ed. John Wiley & Sons, 2021.

- [8] M. Alexandersson, “System identification and prediction models in ship dynamics: From roll damping to manoeuvring,” Ph.D. dissertation, Chalmers University of Technology, Gothenburg, Sweden, 2025, doctoral dissertation.
- [9] H. Abrougui, S. Nejim, and H. Dallagi, “Nonlinear low-level controller design for an autonomous surface vehicle,” in *2023 IEEE International Conference on Advanced Systems and Emergent Technologies (IC_ASET)*, 2023, pp. 1–6.
- [10] H. Xu and C. G. Soares, “Review of system identification for manoeuvring modelling of marine surface ships,” *Journal of Marine Science and Application*, vol. 24, no. 3, pp. 459–478, 2025. [Online]. Available: <https://doi.org/10.1007/s11804-025-00681-w>
- [11] F.-A. Fortin, F.-M. De Rainville, M.-A. Gardner, M. Parizeau, and C. Gagné, “DEAP: Evolutionary algorithms made easy,” *Journal of Machine Learning Research*, vol. 13, pp. 2171–2175, jul 2012.
- [12] Y. Ma, J. Ren, and X. Hu, “Parametric identification of ship abkowitz maneuvering model using extended kalman filter,” in *2024 International Annual Conference on Complex Systems and Intelligent Science (CSIS-IAC)*. IEEE, 2024. [Online]. Available: <https://ieeexplore.ieee.org/>
- [13] L. Abualigah, “Particle swarm optimization: Advances, applications, and experimental insights,” *Computers, Materials and Continua*, vol. 82, no. 2, pp. 1539–1592, 2025. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1546221825000852>
- [14] P. E. Hart, N. J. Nilsson, and B. Raphael, “A formal basis for the heuristic determination of minimum cost paths,” *IEEE Transactions on Systems Science and Cybernetics*, vol. 4, no. 2, pp. 100–107, 1968.
- [15] C. De Boor, *A Practical Guide to Splines*, ser. Applied Mathematical Sciences. Springer New York, 1978. [Online]. Available: <https://books.google.se/books?id=mZMQAQAAIAAJ>
- [16] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. M. Scokaert, “Constrained model predictive control: Stability and optimality,” *Automatica*, vol. 36, no. 6, pp. 789–814, 2000.
- [17] J. B. Rawlings, D. Q. Mayne, and M. M. Diehl, *Model Predictive Control: Theory, Computation, and Design*, 2nd ed. Madison, WI: Nob Hill Publishing, 2017.
- [18] Y. Workineh, H. Mekonnen, and B. Belew, “Numerical methods for solving second-order initial value problems of ordinary differential equations with euler and runge-kutta fourth-order methods,” *Frontiers in Applied Mathematics and Statistics*, vol. Volume 10 - 2024, 2024. [Online]. Available: <https://www.frontiersin.org/journals/applied-mathematics-and-statistics/articles/10.3389/fams.2024.1360628>

- [19] F. Allgower, R. Findeisen, and Z. K. Nagy, “Nonlinear model predictive control: From theory to application,” *Journal of the Chinese Institute of Chemical Engineers*, vol. 35, no. 3, pp. 299–316, 2004.
- [20] A. Bärlund, J. Linder, H. Feyzmahdavian, M. Lundh, and K. Tervo, “Nonlinear mpc for combined motion control and thrust allocation of ships,” *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 14 698–14 703, 2020, 21st IFAC World Congress. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S2405896320324472>
- [21] P. Sopasakis and E. Fresk. (2025) Optimization engine (open) documentation. Accessed: 2026-03-10. [Online]. Available: <https://alphaville.github.io/optimization-engine/>
- [22] L. Stella, A. Themelis, P. Sopasakis, and P. Patrinos, “A simple and efficient algorithm for nonlinear model predictive control,” in *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, 2017, pp. 1939–1944.
- [23] O. Härkegård, “Backstepping and control allocation with applications to flight control,” 2003. [Online]. Available: <https://api.semanticscholar.org/CorpusID:108150482>
- [24] P. A. Servidia, “Control allocation for gimballed/fixed thrusters,” *Acta Astronautica*, vol. 66, no. 3-4, p. 587 – 594, 2010, cited by: 30. [Online]. Available: <https://www.scopus.com/inward/record.uri?eid=2-s2.0-70449521190&doi=10.1016%2fj.actaastro.2009.07.023&partnerID=40&md5=a6192968874dfad1a17136e33bd7e1b7>
- [25] T. I. Fossen and T. A. Johansen, “A survey of control allocation methods for ships and underwater vehicles,” in *2006 14th Mediterranean Conference on Control and Automation*, 2006, pp. 1–6.
- [26] M. Alexandersson, W. Mao, and J. W. Ringsberg, “System identification of vessel manoeuvring models,” *Ocean Engineering*, vol. 266, p. 113218, Dec. 2022. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0029801822032179>

DEPARTMENT OF MECHANICS AND MARITIME SCIENCES
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden
www.chalmers.se



CHALMERS
UNIVERSITY OF TECHNOLOGY