



A Geographically Anchored MMG-Based Ship Maneuvering Simulator for Comparative Control Testing

Ahmad Irham Jambak¹, Mostafa Abdelkhalek², Ismail Bayezit², Omer Kemal Kinaci³

¹Istanbul Technical University Faculty of Mechanical Engineering, Department of Mechatronics Engineering, İstanbul, Türkiye

²Istanbul Technical University Faculty of Aeronautics and Astronautics, Department of Aeronautical Engineering, İstanbul, Türkiye

³Istanbul Technical University Faculty of Naval Architecture and Ocean Engineering, Department of Naval Architecture and Ocean Engineering, İstanbul, Türkiye

To cite this article: A. I. Jambak, M. Abdelkhalek, I. Bayezit, and O. K. Kinaci. A geographically anchored MMG-based ship maneuvering simulator for comparative control testing. *J Nav Architect Mar Technol*. 2026;228:92-116.

Received: 06.06.2026 - **Revision Requested:** 10.08.2026 - **Last Revision Received:** 19.08.2026 - **Accepted:** 12.09.2026 - **Publication Date:** 20.10.2026

Abstract

Reliable evaluation of ship-maneuvering controllers requires more than a dynamically plausible vessel trajectory. For example the maneuvering model, geographically defined scenario, controller interface, and performance evidence must be connected through a reproducible workflow. This study develops a geographically referenced simulator that integrates a Maneuvering Modeling Group (MMG) representation of the full-scale and L7 KVLCC2 model with waypoint guidance, heading and speed control, synchronized data logging, and post-run assessment for an archived route segment in the İstanbul Strait. A dual-loop proportional-integral-derivative (PID) controller, a fuzzy-logic controller, and a neural-network controller are implemented through a common heading-control interface. The recorded signals are evaluated using peak overshoot, median settling time, heading root-mean-square error, rudder-saturation ratio, and maximum speed-loss ratio, with user-defined PASS/WARN/FAIL thresholds. Model-scale and full-scale cases demonstrate the end-to-end route, control, MMG state-propagation, visualization, and diagnostic chain and expose controller-dependent differences in transient response and actuator use. The main contribution is therefore a traceable experimental environment for implementing and examining alternative ship-control strategies within geographically meaningful scenarios.

Keywords: Ship maneuvering simulator, MMG model, comparative control testing, waypoint guidance, fuzzy control, neural network control

1. Introduction

Reliable ship maneuvering is central to safe and efficient operation in ports, narrow channels, and congested waterways, where small heading errors or delayed actuator responses can compromise route keeping and increase rudder activity and speed loss. The problem is consequential for a transport system that carries more than 80% of international trade by volume [1]. The broader economic role of maritime transport is also extensively discussed in maritime economics literature [2]. Additionally, shipping also accounted for 2.89% of global anthropogenic CO₂ emissions in 2018 [3]. Although

a simulator alone cannot establish an environmental benefit, this scale motivates methods that support safer and more efficient controller development. From a control perspective, the difficulty arises from nonlinear, strongly coupled surge-sway-yaw dynamics, actuator constraints, and environmental loads. Controller development must therefore be supported by evidence that is physically meaningful, repeatable, and directly related to operational performance.

Physical model experiments and sea trials are still essential for validating maneuvering predictions. However, their high cost, safety requirements, and limited repeatability make

Address for Correspondence: Assoc. Prof. Dr. Ismail Bayezit, İstanbul Technical University Faculty of Aeronautics and Astronautics, Department of Aeronautical Engineering, İstanbul, Türkiye

E-mail: bayezit@itu.edu.tr

ORCID ID: orcid.org/0000-0001-9345-5108



them impractical at every stage of controller development. Numerical simulation offers a useful intermediate step, allowing control laws to be implemented, tested under challenging conditions, and compared before moving to physical trials [4]. Nevertheless, the strength of simulation-based evidence depends on the accuracy of the maneuvering equations, hydrodynamic coefficients, numerical solver, actuator model, and selected scenarios. Simulation should therefore be viewed as a tool for guiding and reducing the need for physical experiments rather than replacing experimental validation [5].

Among physics-based modeling approaches, the Maneuvering Modeling Group (MMG) framework is particularly suitable for controller-oriented simulation because it separates the contributions of the hull, propeller, and rudder while retaining variables with clear engineering meaning [6,7]. However, model fidelity alone is not enough to ensure a reproducible controller evaluation. A geographically defined route must be converted consistently into the local coordinate frame used by the guidance and control algorithms. Competing controllers must also be tested using equivalent references, constraints, and operating conditions, while reference commands, vessel states, and actuator responses must remain properly synchronized. Without this consistency throughout the simulation process, a trajectory that appears successful offers limited evidence of either prediction accuracy or controller performance.

Previous studies have advanced several elements of this workflow, including MMG-based maneuvering prediction, path-following control, autonomous-navigation test environments, and mission-level simulation [8-12]. However, methodological weakness remains at the interfaces between these elements. When vessel dynamics, geographic visualization, controller implementation, and post-processing are handled in separate tools, it becomes difficult to trace model assumptions and signal definitions across the full workflow. As a result, observed differences in performance may arise from inconsistent scenarios, reference signals, or evaluation metrics rather than from the controllers themselves. An integrated simulator becomes scientifically valuable only when it maintains a clear and traceable connection between route definition, dynamic response, and quantitative evaluation.

Against this background, this study develops and evaluates an MMG simulation environment designed to support repeatable controller implementation and post-simulation assessment. The platform combines the full-scale and L7 KVLCC2 vessel models with waypoint-based guidance and a geographic representation of an archived route through the İstanbul Strait. It also maintains a consistent exchange of heading references, vessel states, and actuator commands

throughout the simulation workflow. Manual control, dual-loop proportional-integral-derivative (PID) heading and speed control, fuzzy-logic (FL) heading control, and neural-network (NN) heading control are implemented as representative case studies.

This study examines whether route definition, maneuvering dynamics, controller implementation, and performance evaluation can be combined within a single traceable workflow without obscuring the assumptions that shape the results. To investigate this issue, the simulator records synchronized reference commands, vessel states, and actuator responses throughout each run. Completed simulations are then assessed using measures of transient response, tracking error, rudder activity, and speed loss. The same signal and data-processing structure is applied to all controller modes, allowing their performance to be compared within a consistent experimental framework.

This work does not propose a new MMG force formulation or controller-synthesis method. Instead, its contribution is the organization of the simulation chain into a coherent and traceable experimental workflow. This distinction is important because the scientific value of the platform is determined not by the number of software functions it includes, but by the extent to which route, model, controller, and diagnostic information remain connected and interpretable throughout each simulation run.

Within this scope, the study makes three related contributions:

- (i) A unified architecture that ensures consistent data exchange among geographic route definition, waypoint guidance, controller calculations, and MMG state propagation.
- (ii) An interchangeable controller interface that provides a common structure for reference signals, vessel states, actuator commands, and logged data across PID, FL, and NN implementations.
- (iii) A traceable assessment layer that connects geographic route execution with heading transients, root-mean-square tracking error, rudder saturation, speed loss, and user-defined diagnostic thresholds.

Full-scale and L7 KVLCC2 model cases are presented to demonstrate the complete workflow and show how different controller configurations lead to different recorded responses. These cases are not intended as a controlled benchmark and therefore cannot be used to establish hydrodynamic prediction accuracy or controller superiority. The study is consequently positioned as a contribution to simulator workflow development. Stronger performance claims would require benchmark maneuver validation, standardized controller-tuning procedures, environmental-disturbance scenarios, and repeated quantitative experiments. The remainder of the paper reviews related studies, describes the simulator architecture and MMG formulation, presents the control and

assessment modules, examines the demonstration cases, and discusses the main limitations and future validation needs.

2. Related Work and Research Gap

2.1. Ship Maneuvering Simulators for Control Testing

Ship maneuvering simulators have become an important tool for the development and validation of maritime control systems. Compared with physical sea trials, simulation-based testing provides a repeatable and low-risk environment in which different control strategies can be evaluated before implementation on real vessels. This is especially important for autonomous and semi-autonomous ship applications, where controller behavior must be examined under various operating conditions, including waypoint tracking, restricted-water navigation, and maneuvering in the presence of other vessels.

Recent studies increasingly treat maritime simulators as engineering platforms for autonomy development rather than solely as training tools [8,9]. A preliminary MMG-based ship simulator integrating autonomous PID control, manual operation, geographic visualization, and real-time vessel-state monitoring was presented in [13]. Dai et al. [10] developed an immersive virtual reality-based testing method and performance indicator system for autonomous navigation. Hinostroza and Lekkas [11] combined temporal mission planning with a guidance, navigation, and control framework in a realistic marine environment. Together, these studies highlight the importance of linking scenario definition, vessel dynamics, autonomy functions, and performance evaluation within a consistent simulation framework.

The review by Xu and Guedes Soares [12] organizes maritime autonomous surface-ship path-following research around guidance, control, vessel dynamics, stability, and environmental disturbances. For simulator-based assessment, these elements imply that reference generation, tracking error, control action, and maneuvering-model assumptions should be recorded together with the vessel trajectory. Robust-control studies have additionally considered the effect of hydrodynamic-parameter uncertainty on nonlinear surface-vessel control [14].

2.2. Data-Driven Modeling and Maneuver Prediction

In addition to model-based simulation, data-driven techniques have been investigated for ship maneuvering prediction and modeling. Long Short-Term Memory networks have been applied to identify and predict ship maneuvering motion by learning nonlinear time-dependent behavior from maneuvering data [15]. Sensor-data-based identification has also been combined with modular autopilot instrumentation and PID heading control [16]. Artificial Neural Networks have also been widely reviewed and applied in maritime

applications, including modeling, prediction, and decision-making tasks [17]. These methods are useful because ship maneuvering dynamics are nonlinear and can be difficult to describe accurately under all operating conditions using only analytical models.

Other studies focus on maneuvering prediction, navigation decision-making, or route generation rather than on simulator validation itself [18]. Their relevance to the present study is therefore indirect. Although they help identify the information that a simulation platform should provide, they do not verify the accuracy of the MMG implementation or ensure that controller comparisons are conducted under equivalent conditions.

MMG-based simulation is particularly useful when controller assessment requires physically interpretable surge, sway, yaw, rudder, and propeller variables. Mitsuyuki et al. [19] demonstrated uncertainty-aware identification of MMG hydrodynamic maneuvering coefficients from observed motion and control data. Liu et al. [20] combined an MMG model with event-triggered predictive path-following control and evaluated tracking performance together with steering activity. These studies motivate an assessment structure that links model assumptions, controller computation, and actuator-level measurements.

Taken together, recent studies address model identification, guidance and control, autonomous-navigation testing, and mission planning at different levels of the simulation chain. The present study focuses on their implementation interface by combining geographic route execution, MMG state propagation, interchangeable heading-control modules, synchronized logging of vessel and actuator variables, and post-run diagnostic calculation in one simulator. The contribution is the integrated and traceable assessment process rather than a new MMG formulation or control law.

2.3. Route Optimization and Maritime Decision Support

Another relevant research direction is maritime route optimization and decision support. Route optimization studies aim to improve navigation safety and efficiency while considering geographical constraints, traffic density, and port-area characteristics. For example, ship path optimization methods that account for geographical traffic characteristics have been proposed to improve maritime port safety [21]. More recently, model-based predictive path planning has incorporated MMG vessel dynamics directly into constrained-environment path generation [22]. Such studies are relevant to maneuvering simulator development because route planning and path-following performance are strongly connected to the vessel's control capability. Model-based navigation and control have also been investigated for tankship operation in narrow waterways [23].

Nevertheless, many route optimization and decision-support studies focus mainly on path generation or safety

evaluation. In contrast, a control-testing simulator must also evaluate whether a particular vessel model and controller can physically follow the generated path with acceptable heading error, overshoot, settling behavior, actuator usage, and speed response. This distinction is important because a geometrically feasible path may still be difficult for a real ship to follow due to hydrodynamic limitations and actuator constraints. Feasibility can also depend on the propulsion mechanism. For autonomous sailing vehicles, geometrically safe avoidance headings may become unsailable when they enter the wind no-go zone [24].

2.4. Industrial Simulator and Navigation Platforms

Industrial developers have also made significant progress in maritime simulators, navigation-support systems, and autonomous-vessel technologies. Kongsberg has developed advanced maritime simulation platforms, including systems used for training and the evaluation of autonomous ship operations [25]. Furuno has introduced navigation-support technologies such as augmented-reality navigation and autonomous marine solutions, where radar, AIS, camera information, and visualization tools are integrated to improve situational awareness [26-28]. JRC has focused on

simulator systems that support maritime pilot training and environmental-condition representation [29].

Other industrial platforms emphasize training, traffic simulation, and operational decision support. Wärtsilä provides navigation simulation systems with functions for user-performance evaluation during training scenarios [30]. Transas has contributed to maritime traffic simulation and full-mission bridge simulators, supporting traffic management and risk-analysis applications [31]. Siemens has investigated autonomous navigation and intelligent co-pilot concepts for supporting operational decisions [32]. Rolls-Royce has also worked on autonomous ship concepts and simulation-supported development for future remote-controlled and autonomous vessels [33].

Commercial platforms demonstrate the practical importance of maritime simulation, training, and navigation support. However, publicly available product material generally does not disclose model equations, controller source code, parameter files, or reproducible benchmark data. Table 1 therefore compares documented functional emphasis rather than making claims about unreported cost, performance, or internal architecture.

Table 1. Comparison of selected industrial maritime simulator and navigation-support platforms.

Platform	Main focus	Relevant capability	Limitation from control-testing perspective	Relevance to the present study
Kongsberg [25]	Maritime simulation and decision support	Supports complex maneuvering scenarios and autonomous ship operation evaluation	Public information does not expose model equations, controller source code, or reproducible benchmark data.	Shows the importance of high-fidelity simulation for maneuvering and control-system evaluation
Furuno [26-28]	Navigation visualization and situational awareness	Integrates navigation information, camera-based visualization, radar, AIS, and 3D situational display	Public information emphasizes navigation assistance; controller-level extensibility is not documented.	Motivates the integration of map-based visualization and real-time navigation feedback in the simulator
JRC [29]	Simulator-based maritime training and environmental representation	Provides realistic training environments and supports the representation of environmental conditions	Public information emphasizes training; controller-level extensibility	Highlights the need to include realistic operating conditions while maintaining control-test flexibility
Wärtsilä [30]	Navigation training and user-performance assessment	Evaluates user decisions and supports targeted training improvement	Public information emphasizes training and assessment; research access and algorithm modification are not documented.	Motivates the use of post-run analysis and performance assessment after simulation
Transas [31]	Maritime traffic simulation and full-mission bridge simulation	Supports traffic management, risk analysis, and complex maritime scenario generation	Public information emphasizes operational training and traffic simulation; controller source-code access is not documented.	Shows the value of traffic-aware and scenario-based simulation for future simulator extension
Siemens [32]	Autonomous navigation and intelligent co-pilot systems	Supports route generation and operational decision-making under different conditions	Public information emphasizes operational decision support; research-oriented controller modification is not documented.	Indicates the relevance of decision-support functions in autonomous ship control testing
Rolls-Royce [33]	Autonomous ship concepts and simulation-supported development	Focuses on environmental awareness, remote operation, and autonomous-vessel development	Public information describes proprietary autonomous-vessel concepts; reproducible controller-level access is not documented.	Provides industrial motivation for developing transparent and user-configurable, simulators

On this basis, the study examines whether a geographically referenced MMG simulator can provide a coherent path from controller implementation to post-run maneuver assessment. The platform is assessed in terms of:

- (i) Integration of guidance, control, vessel dynamics, visualization, and logging.
- (ii) Execution of interchangeable heading-control implementations.
- (iii) Calculation of maneuver-quality indicators from a saved route run.

Controller plots are used to demonstrate platform operation, whereas quantitative interpretation is restricted to cases for which compatible raw logs are available.

3. Simulator Framework

3.1. Overall Architecture

The simulator is structured as an interactive control-development environment composed of five tightly

connected layers: waypoint guidance, heading and speed control, MMG-based maneuvering dynamics, geographic conversion and visualization, and post-run data analytics. Figure 1 summarizes this organization.

The guidance layer interprets user-defined waypoints and transforms them into sequential maneuvering objectives. The control layer generates rudder and propulsion commands either autonomously or through manual intervention. The plant layer propagates vessel motion using the MMG model. The visualization layer converts Cartesian states back to geographic coordinates for map-based display. Finally, simulation logs are stored for offline analysis, replay, and automatic quality assessment. More detailed descriptions of each layer are presented in Table 2.

3.2. MMG-Based Maneuvering Model

The ship model employed in this study adheres to the standardized MMG (Maneuvering Mathematical Group)

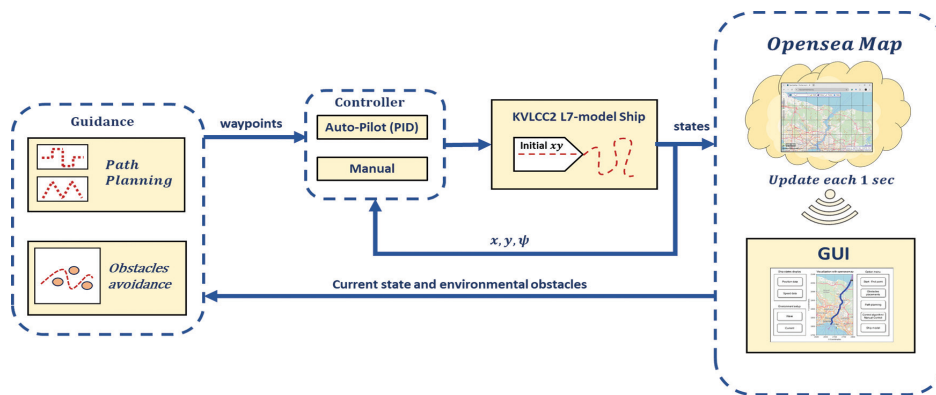


Figure 1. Overall ship-simulator software model.

PID: Proportional-integral-derivative, GUI: Graphical User Interface

Table 2. Functional layers of the simulator architecture.

Layer	Main role	Typical inputs	Main outputs
Input and initialization layer	Defines route, initial conditions, vessel parameters, actuator limits, disturbance settings, and simulation settings.	Waypoint CSV file, ship parameters, initial state, controller parameters, simulation time-step/update settings.	Validated initial configuration and local Cartesian waypoint set.
Guidance and reference layer	Selects the active waypoint and computes the reference heading or route-following command.	Current vessel position, waypoint list, waypoint-switching rule.	Reference heading, active waypoint index, path-tracking information.
Control layer	Converts heading and speed errors into rudder and propeller commands using the selected controller mode.	Reference heading, measured heading, reference speed, measured speed, controller parameters.	Rudder command, propeller-speed command, selected control mode.
Ship-dynamics layer	Propagates the vessel states using the maneuvering model and actuator commands.	Rudder angle, propeller speed, hydrodynamic coefficients, vessel state.	Surge speed, sway motion, yaw rate, position, heading.
Visualization and analysis layer	Displays the run, stores simulation data, plots selected variables, and evaluates maneuver performance after the run.	State histories, control histories, map data, performance thresholds.	Map trajectory, time histories, exported data, post-run assessment report.

ship modeling framework [6,7]. Ship maneuvering in still water involves three primary motions: surge, sway, and yaw. These can be described by the following equations of motion:

$$(m + m_x)\ddot{u} - (m + m_y)\dot{v}_m r - x_G m r^2 = X \quad (1a)$$

$$(m + m_y)\dot{v}_m + (m + m_x)ur + x_G m \dot{r} = Y \quad (1b)$$

$$(I_{zG} + x_G^2 m + J_z)\dot{r} + x_G m(\dot{v}_m + ur) = N_m \quad (1c)$$

where u , v , and r are the surge velocity, sway velocity, and yaw rate, respectively. For practical purposes, the coupling terms involving added mass in v_m and r are neglected. The total hydrodynamic forces can be decomposed as follows:

$$X = X_H + X_R + X_P \quad (2a)$$

$$Y = Y_H + Y_R \quad (2b)$$

$$N_m = N_H + N_R \quad (2c)$$

Here, the subscripts H , R , and P refer to contributions from the hull, rudder, and propeller, respectively. The hydrodynamic forces on the hull, X_H , Y_H , and N_H can be modeled as:

$$X_H = 0.5\rho L_{pp} d U^2 X'_H(v'_m, r') \quad (3a)$$

$$Y_H = 0.5\rho L_{pp} d U^2 Y'_H(v'_m, r') \quad (3b)$$

$$N_H = 0.5\rho L_{pp}^2 d U^2 N'_H(v'_m, r') \quad (3c)$$

where v'_m and r' are the non-dimensional lateral velocity and yaw rate, defined as $\frac{v_m}{U}$ and $r \frac{L_{pp}}{U}$. These coefficients are expanded into polynomial forms:

$$X'_H(v'_m, r') = -R'_0 + X_{vv} v_m'^2 + X_{vr} v'_m r' + X_{rr} r'^2 + X_{vvv} v_m'^4 \quad (4a)$$

$$Y'_H(v'_m, r') = Y_v v'_m + Y_R r' + Y_{vvv} v_m'^3 + Y_{vvr} v_m'^2 r' + Y_{vrr} v'_m r'^2 + Y_{rrr} r'^3 \quad (4b)$$

$$N'_H(v'_m, r') = N_v v'_m + N_R r' + N_{vvv} v_m'^3 + N_{vvr} v_m'^2 r' + N_{vrr} v'_m r'^2 + N_{rrr} r'^3 \quad (4c)$$

The surge force generated by the propeller is expressed as:

$$X_P = (1 - t_p)T \quad (5)$$

where the propeller thrust T is given by:

$$T = \rho n_p^2 D_p^4 K_T(J_p) \quad (6)$$

with the thrust coefficient $K_T(J_p)$ modeled as a second-order polynomial:

$$K_T(J_p) = k_2 J_p^2 + k_1 J_p + k_0 \quad (7)$$

and the advance ratio defined as:

$$J_p = \frac{u(1 - w_p)}{n_p D_p} \quad (8)$$

The wake fraction w_p depends on the flow conditions and is approximated by:

$$\frac{1 - w_p}{1 - w_{p0}} = 1 + (1 - \cos^2 \beta_p)(1 - |\beta_p|) \quad (9)$$

where $\beta_p = \beta - x'_p r'$ is the inflow angle at the propeller. The forces and moment due to steering are given by:

$$X_R = -(1 - t_R) F_N \sin \delta \quad (10a)$$

$$Y_R = -(1 + a_H) F_N \cos \delta \quad (10b)$$

$$N_R = -(x_R + a_H x_H) F_N \cos \delta \quad (10c)$$

where the rudder normal force F_N is defined as:

$$F_N = 0.5\rho A_R U_R^2 f_a \sin \alpha_R \quad (11)$$

The rudder inflow velocity U_R and inflow angle α_R are given by:

$$U_R = \sqrt{u_R^2 + v_R^2} \quad (12)$$

$$\alpha_R = \delta - \arctan \frac{v_R}{u_R} \approx \delta - \frac{v_R}{u_R} \quad (13)$$

The lateral velocity at the rudder, v_R , is influenced by flow straightening and is modeled as:

$$v_R = U \gamma_R \beta_R \quad (14)$$

where $\beta_R = \beta - l_R r'$ and γ_R accounts for the flow alignment effect.

The physical and hydrodynamic parameters used for the full-scale and L7 KVLCC2 models were obtained from [6,7]. The present implementation was not independently validated against experimental or published turning-circle and zig-zag benchmark results. Therefore, the simulations reported here are not intended to establish maneuvering-prediction accuracy.

3.3. Waypoint Guidance and Reference Generation

Although the user interacts with waypoints in latitude and longitude, control calculations are performed in a local Cartesian frame for numerical simplicity. Let (x_i^{wp}, y_i^{wp}) denote the current target waypoint and (x, y) the ship position. The commanded heading is then obtained from

$$\psi_{ref} = \text{atan2}\left(\frac{x_{ref,i}}{y_{ref,i}}\right) \quad (15)$$

$$\psi_{error} = \psi_{ref} - \psi_{cur} \quad (16)$$

where ψ is the current vessel heading. Figure 2 shows the resulting closed-loop information flow.

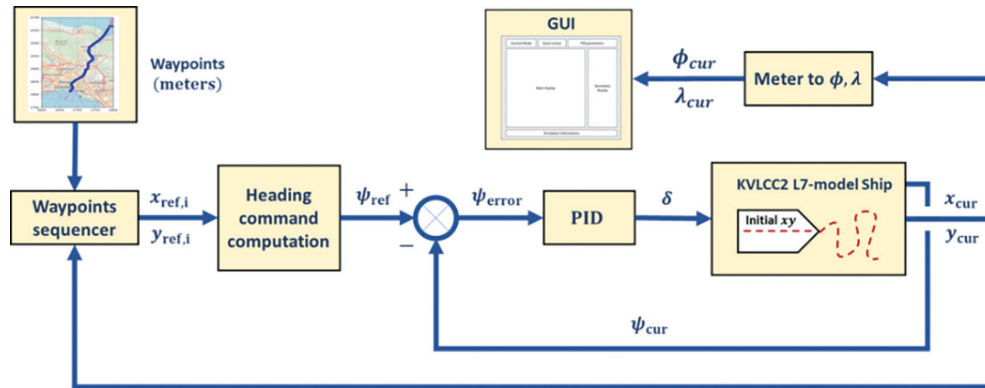


Figure 2. Closed-loop waypoint-following structure linking geographic waypoints, heading-command generation, control action, MMG state propagation, and GUI display.

MMG: Maneuvering Modeling Group

4. Simulator Product Layout and Interaction Design

4.1. Product-Oriented Design Philosophy

The simulator exposes the maneuvering workflow through an operator-facing interface rather than running the equations as an isolated numerical backend. Parameter initialization, controller selection, real-time visualization, signal logging, and post-run analysis are therefore treated as connected stages of a controller-development process.

The implemented software uses a tabbed desktop interface for configuration, execution, inspection, and evaluation. Table 3 summarizes the capability groups and their role in the workflow.

4.2. Tab-Based Workspace Architecture

The interface is divided into four functional tabs. Table 4 summarizes the responsibilities of each tab. The tabbed

structure separates scenario initialization, run-time operation, data inspection, and post-run assessment. The implementation is described as following a Model-View-Controller organization with asynchronous controller execution and coordinate-conversion services.

4.2.1. Initialization Tab

According to the project reports, the initialization tab was designed around four core responsibilities:

- waypoint CSV import and automatic geographic calculations,
- ship-model and parameter loading,
- dynamic validation of parameters,
- multi-column organization of configuration fields.

The waypoint-import path is especially important because the simulator operates visually in geographic space but computes control actions in a local Cartesian frame. Once

Table 3. Integrated capability groups of the simulator product.		
Capability group	Main implemented elements	Product value
Runtime simulation	Istanbul Strait route visualization, live telemetry, autonomous/manual operation, and real-time state updates	Makes maneuvering behavior spatially and dynamically interpretable while the simulation is running.
Controller workspace	PID-based heading and speed control loops, FL configuration, and neural-network control integration	Enables the simulator to function as a controller-development environment rather than only as a fixed playback tool.
Parameter management	Ship-model selection, editable MMG and physical parameters, disturbance settings, and initial-condition fields	Supports reusable scenario preparation under different vessel models and operating assumptions.
Data analysis	Log replay, axis selection, overlay plots, save/export actions, and embedded workspace panels	Preserves engineering evidence within the same application after the simulation run is complete.
Diagnostic review	Metric extraction, PASS/WARN/FAIL logic, radar-chart summaries, comparison actions, and recommendation text	Converts raw simulation traces into structured assessments of maneuvering quality.

Table 4. Functional organization of the simulator workspace.

Tab	Main objective	Principal widgets and controls	Typical outputs
Initialization	Configures the scenario and plant model before execution.	Waypoint CSV import, ship-model selection, editable physical parameters, parameter-validation fields, and multi-column setup panels.	Loaded route, validated ship parameters, and local Cartesian reference data.
Simulation	Executes autonomous or manual navigation in real time.	Controller mode selector, zoom controls, map canvas, secondary display, live plots, pause/restart/stop logic, and controller-parameter panel.	Ship motion on the Istanbul Strait map, instantaneous telemetry, and live heading, rudder, and speed traces.
Data analysis	Inspects completed runs without rerunning the plant model.	Dataset loader, workspace list, axis selectors, plot mode selector, plot/hold/clear/save actions, and table export.	Time-series plots, variable-versus-variable plots, overlay comparisons, and exported figures and tables.
Post run analyzer	Converts simulation logs into engineering judgments.	Metric threshold editor, comparison source selector, radar chart, report window, save-report action, and run-analysis trigger.	PASS/WARN/FAIL assessment, segment-level diagnostics, and controller-tuning recommendations.

a CSV route containing latitude and longitude values is loaded, the software transforms the route relative to the selected starting point, thereby preparing the waypoints for the guidance and control layers. The reports emphasize that this transformation is automatic, which reduces operator workload and helps maintain consistency between the visual map and the control-side geometry.

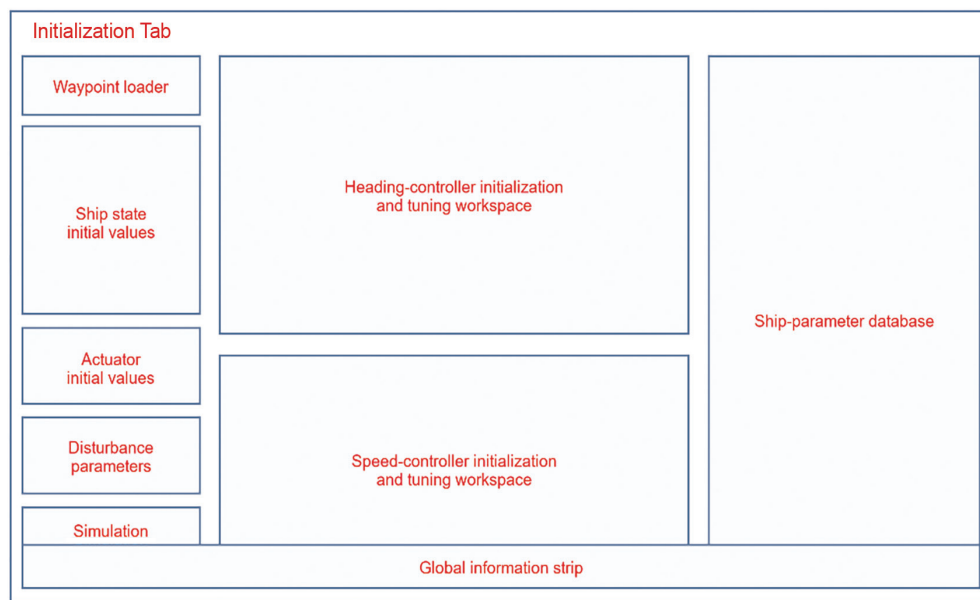
The ship-model panel plays an equally important role. It allows the user to inject or edit dimensions, mass properties, and maneuvering coefficients directly from the Graphical User Interface (GUI). In a product paper this is worth emphasizing because it moves the simulator away from being a hard-coded demonstrator and toward being a reusable vessel-testing platform.

Figure 3 summarizes the initialization workspace, including route loading, initial-state specification, disturbance configuration, controller setup, and vessel-parameter editing. Table 5 lists the corresponding functional groups.

4.2.2. Simulation Tab

The simulation tab is the visual core of the simulator. Figure 4 summarizes the principal regions of the runtime window schematically.

The top band contains run-time controls, the center-left region displays the geographic trajectory, the right column presents context and instrumentation views, and the lower status strip reports numerical telemetry. Tables 6 and 7 summarize these regions and variables.



Schematic layout used to explain setup-space organization before showing runtime GUI views.

Figure 3. Schematic layout of the initialization tab showing route loading, initial-state panels, controller-initialization workspaces, the ship-parameter database, and the global information strip.

Table 5. Functional clustering inside the initialization tab in the implemented GUI.

Cluster	Main contents	Product purpose
Waypoint loader	CSV path display, file-open action, and total waypoint count.	Connects the navigation scenario to the rest of the setup process and makes the route provenance explicit.
State and disturbance panels	Initial latitude, longitude, heading, surge, sway, and yaw states, actuator initial values, wave parameters, and simulation-step settings.	Defines the operating point from which all later controller behavior should be interpreted.
Heading-controller workspace	PID tuning fields, test setup, response plotting area, and controller-tab hierarchy.	Allows the operator to tune and inspect heading-control behavior before or during scenario definition.
Speed-controller workspace	Speed PID tuning fields, manual test setup, and response plotting area.	Separates propulsion regulation from heading regulation while keeping both visible in the same engineering screen.
Ship-parameter database	Ship-model selector and editable hydrodynamic, geometric, and actuator parameters.	Turns the simulator into a reusable vessel-testing platform rather than a fixed one-ship demonstrator.
Information strip	Global status and information line.	Communicates the active application state without requiring additional dialogs.

GUI: Graphical User Interface, PID: Proportional-integral-derivative

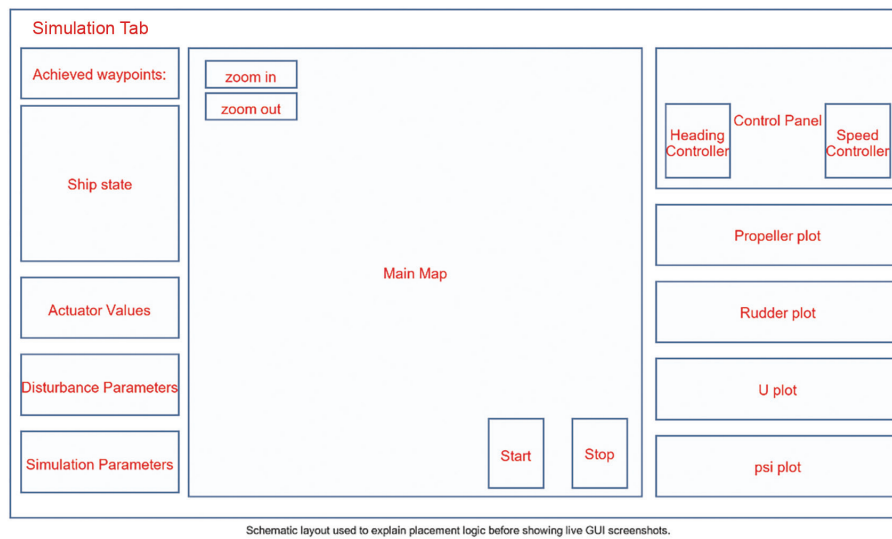


Figure 4. Schematic layout of the simulation tab showing the placement of the main map, left-side state panels, right-side control and plot regions, and run-control buttons.

Table 6. Runtime layout of the simulation tab.

Region	Placement	Contents	User benefit
Control mode selector	Top-left	Active control-mode indicator and autonomous/manual switching logic.	Makes control authority explicit and prevents mode ambiguity during testing.
Zoom controls	Top-center-left	Zoom-in and zoom-out buttons for the main map.	Allows rapid context changes during dense waypoint inspection.
Controller parameter strip	Top-center-right	Gain-entry boxes and apply action for the active controller.	Supports on-the-fly tuning during experimental runs.
Main map display	Center-left, dominant area	OpenSeaMap background, ship marker, route, waypoint geometry, and trajectory history.	Provides the primary situational-awareness view and makes maneuvering behavior spatially interpretable.
Secondary display column	Right side	Overview map and time-history plots of heading, rudder angle, and ship speed.	Provides compact local and global context without leaving the live simulation run.
Status bar	Bottom, full width	Simulation time, propeller speed, rudder angle, ship speed, latitude, longitude, heading, and waypoint count.	Consolidates machine-state information for quick reading and logging.

Table 7. Primary real-time variables exposed through the simulation-tab status strip.

Field	Unit or type	Engineering interpretation
Simulation time	s	Establishes event chronology and allows the operator to relate turning behavior to control action and waypoint switching.
Propeller speed (n_p)	rps	Indicates commanded propulsion effort and supports interpretation of surge-speed transients.
Rudder angle (δ)	deg	Shows immediate steering demand and is important for identifying controller saturation or oscillatory steering.
Ship speed (U)	m/s	Summarizes forward-motion efficiency and reveals maneuver-induced speed loss.
Latitude and longitude	deg	Keep the live trajectory anchored to the geographic context of the Istanbul Strait map.
Heading (ψ)	deg	Gives the vessel's instantaneous orientation for comparison with the reference-heading evolution.
Waypoint count/active route state	integer/state label	Informs the operator which part of the mission is currently being executed and whether waypoint advancement is behaving correctly.

During a run, the map displays the route, waypoints, vessel marker, and trajectory, while adjacent plots and telemetry panels expose selected state and actuator variables. Pause, restart, and stop functions support interactive testing.

4.2.3. Data Analysis Tab

The data-analysis tab loads completed simulation datasets and supports signal selection, overlay plots, table inspection, figure export, and numerical-data export (Table 8).

Keeping these operations within the simulator allows reference, state, and actuator histories to be inspected without rerunning the model. The workflow places dataset management on the left, signal selection at the top, numerical

inspection in the center, and plots in the lower workspace. Figure 5 summarizes the data-analysis workspace and separates run management, plotting controls, numerical data, and the plot canvas.

4.2.4. Post-run Analyzer Tab

The post-run analyzer organizes completed-run assessment. The arrangement allows the user to inspect the detected context and thresholds before comparing the radar chart with the textual metric summary and recommendations. Figure 6 summarizes the post-run analyzer and separates context detection, action controls, threshold editing, radar-chart visualization, and the textual report.

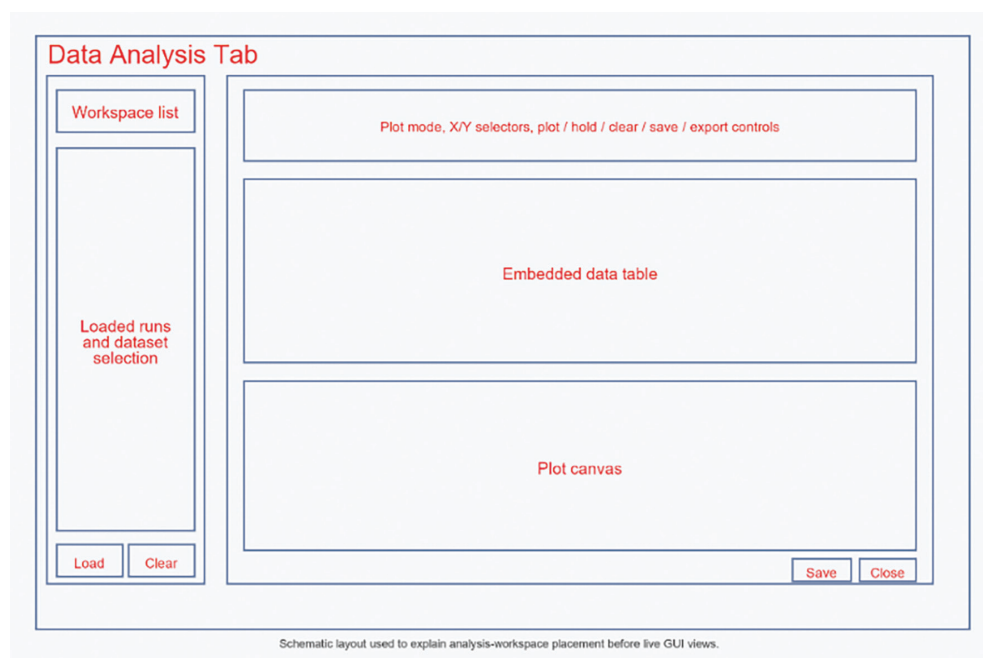


Figure 5. Schematic layout of the data-analysis tab showing the workspace list, plot-control strip, embedded data table, plot canvas, and export-oriented utility buttons.

GUI: Graphical User Interface

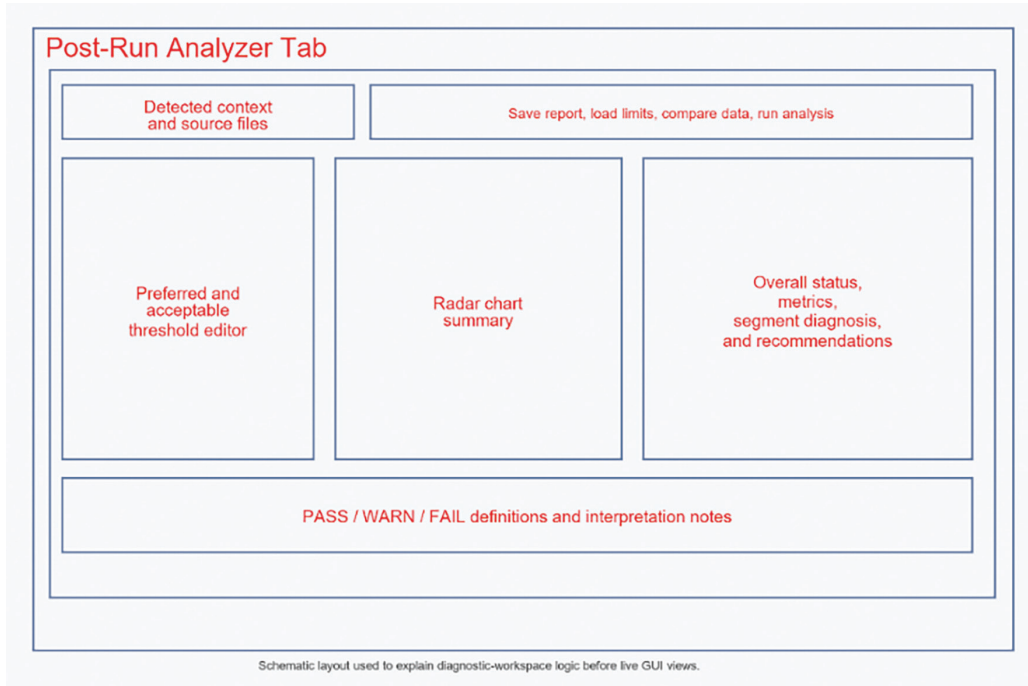


Figure 6. Schematic layout of the post-run analyzer tab showing context detection, analysis actions, threshold editing, the radar-chart panel, the textual diagnostic report, and the PASS/WARN/FAIL interpretation strip.

Table 8. Interaction logic of the data-analysis tab.

Interface element	Placement	Function in the workflow
Workspace list	Left column	Stores one or more completed runs so that the user can switch among experiments without relaunching the simulation.
Load/clear workspace actions	Upper-left and lower-left	Adds new datasets to the session and resets the analysis workspace when a fresh comparison cycle is needed.
Plot-mode selector	Upper control band	Switches between time-history plotting and variable-versus-variable interpretation.
X/Y axis selectors	Upper control band	Exposes simulation variables directly to the plotting engine and supports fast custom comparisons.
Plot, hold, clear, and save actions	Upper control band	Creates overlays, resets the current figure, and preserves publication-ready or report-ready plots.
Save table data action	Upper control band	Exports numerical logs for use in external spreadsheets or additional statistical analysis.
Embedded data table	Central pane	Provides exact sample-level values for debugging, sanity checks, and parameter tracing.
Plot canvas	Lower pane	Produces the visual engineering evidence used to compare runs and interpret controller behavior.

5. Control and Intelligence Modules

5.1. Controller Exposure Through the GUI

The controller modules are presented primarily as replaceable implementations that exercise the simulator workflow. Formal controller synthesis, gain optimization, robustness analysis, and statistically controlled benchmarking are outside the scope of the current evidence. Table 9 summarizes the control-related inputs exposed through the interface.

5.2. Dual-Loop PID Control

Control-oriented ship models have also been used with cascade-control architectures for vessel motion control [34]. PID control is retained as the main reference controller because of its interpretability and tuning transparency. The heading-control command is

$$\delta_{\text{cmd}}(t) = K_{P,\psi} e_{\psi}(t) + K_{I,\psi} \int_0^t e_{\psi}(\tau) d\tau + K_{D,\psi} \dot{e}_{\psi}(t) \quad (17)$$

Table 9. Controller-specific interaction capabilities of the simulator GUI.

Controller model	Exposed user inputs	Main engineering purpose	Reported interface emphasis
PID heading control	Editable (K_p), (K_I), and (K_D) values for the heading loop.	Tunes responsiveness, overshoot, and settling behavior during path following.	Baseline autonomous control and reference benchmark.
PID speed control	Editable (K_p), (K_I), and (K_D) values for the propulsion loop.	Regulates forward speed independently from heading stabilization.	Dual-loop precision maneuvering and speed maintenance.
Fuzzy logic control	Membership-function type, parameter ranges, and fuzzy-rule configuration.	Shapes nonlinear heading response and improves adaptability across operating conditions.	User-configurable linguistic control with smoother corrective behavior after tuning.
Neural network control	Training-data selection, network topology, training execution, and persistence support.	Develops data-driven heading-control policies and compares them against classical designs.	Background training integration and adaptive control experimentation.
Manual mode	Direct rudder commands through the keyboard.	Inspects plant behavior without autopilot authority and supports operator familiarization.	Interactive testing and training support.

subject to actuator saturation, typically within ± 30 deg. Where the speed error is defined as the difference between the reference speed and the measured surge speed. This dual-loop structure allows heading and forward-speed regulation to be studied independently and jointly. The heading and speed loops operate independently, although their responses remain coupled through the MMG vessel dynamics. The parameter configurations used to demonstrate the controller workspace for the PID controller are summarized in Table 10.

The gains reported in Table 10 were selected through limited exploratory simulation to produce bounded and visibly distinguishable responses within the configured actuator limits. For the full-scale heading example, a proportional gain of $K_p=4$ was first retained after it produced a usable closed-loop response. Small integral and derivative terms were then varied to demonstrate how changes in the entered parameters are reflected in the simulated response and subsequent analysis. The corresponding L7 and speed-control configurations were prepared for the same software-demonstration purpose. Therefore, these values should not be interpreted as optimal, generally applicable, or recommended controller gains. Configuration A and B were selected to produce different response characteristics for demonstration of the tuning and analysis functions.

5.3. FL Heading Control

A fuzzy-logic (FL) heading controller is included as a rule-based nonlinear alternative. The implementation maps heading-error information to a bounded rudder command using configurable membership functions, inference rules, and centroid defuzzification. A representative triangular membership function is

$$\mu_{\text{tri}}(x; a, b, c) = \begin{cases} 0 & x \leq a \\ \frac{x-a}{b-a} & a < x \leq b \\ \frac{c-x}{c-b} & b < x < c \\ 0 & x \geq c \end{cases} \quad (18)$$

while the crisp rudder command is obtained by centroid defuzzification,

$$\delta_{\text{FL}} = \frac{\sum_i \mu_i(x) x_i}{\sum_i \mu_i(x)} \quad (19)$$

The interface uses a heading-error range of ± 40 deg and a rudder-command range of ± 30 deg. Five linguistic sets are defined for both the heading-error input and rudder-angle output. The input sets are left large, left small, center, right small, and right large, while the corresponding output sets are hard left, soft left, neutral, soft right, and hard right.

Table 10. Illustrative PID parameter configurations used to demonstrate the controller workspace.

Model and loop	Config.	K_p	K_I, K_D	Model and loop	Config.	K_p	K_I, K_D
Full-scale heading	A	4	0.002, 15	L7 heading	A	20	0.01, 0.001
Full-scale heading	B	4	0.001, 0.001	L7 heading	B	20	0.01, 0.01
Full-scale speed	A	20	0.01, 0.001	L7 speed	A	35	0.3, 0.001
Full-scale speed	B	20	0.01, 0.01	L7 speed	B	20	0.1, 0.001

The implemented rule base maps these sets directly as summarized in Table 11. The GUI also allows the membership-function type and control ranges to be configured. These settings are provided to demonstrate the fuzzy-controller implementation within the simulator rather than an optimized fuzzy-control design.

The fuzzy controller therefore contains five rules, five input linguistic sets, and five output linguistic sets. The heading-error and rudder universes of discourse are $[-40,40]$ deg and $[-30,30]$ deg, respectively. Centroid defuzzification is used, and a 50-point precomputed input-output map with linear interpolation is employed for faster execution during simulation.

5.4. NN Heading Control

The neural-network (NN) controller is introduced as a data-driven alternative for mapping maneuvering error to control action. Training data were generated by repeatedly simulating the ship under constant rudder commands. Each sample covered a 10 s interval, and a dataset of 1000 samples was reported. The training input is the heading discrepancy

$$\Delta\psi = \psi_d - \psi_{\text{init}} \quad (20)$$

where ψ_d is the desired final heading and ψ_{init} is a randomly selected initial heading. The network output is the rudder demand. The model was trained with a mean-squared-error objective,

$$L(\theta) = \frac{1}{N} \sum_{i=1}^N (\delta_i - \hat{\delta}_i)^2 \quad (21)$$

using the Adam optimizer. The generic Adam update can be written as

$$\theta_{k+1} = \theta_k - \alpha \frac{\hat{m}_k}{\sqrt{\hat{v}_k + \varepsilon}} \quad (22)$$

where $\hat{m}_k$ and $\hat{v}_k$ are the bias-corrected first and second moment estimates. The specific NN configurations used for the reported L7 and full-scale demonstrations are summarized in Table 12. Each network uses ReLU activation in the hidden layers and a single linear output neuron representing rudder demand. Adam optimization and a mean-squared-error objective are used for all configurations. Training data consist of 1000 simulated samples with a 10 s maneuver interval per sample, and no explicit validation split is used in the implemented training call.

5.5. Automatic Post-Run Evaluation Logic

The simulator includes an automatic post-run analyzer intended to classify the quality of a completed maneuver. Instead of acting online, this module consumes simulation logs after execution and produces a structured diagnostic outcome. Figure 7 summarizes the logic.

The metrics are defined in a segment-aware manner so that major heading changes can be evaluated separately before being aggregated. Table 13 summarizes the diagnostic measures used by the analyzer and the engineering meaning attached to each on.

6. Simulator Interface and Experimental Results

6.1. Implemented GUI Views

Figures 8-11 show the implemented interface across the four main tabs: initialization, simulation, data analysis, and post-run analysis. Together they demonstrate that the simulator operates as a continuous engineering workflow rather than a single visualization screen. Route preparation, live maneuvering, result inspection, and performance diagnosis are all handled inside the same environment, which makes repeated controller tuning and comparison considerably easier.

Table 11. FL controller structure used in the simulator.

Heading-error set	Rudder-output set	Rule
Left large	Hard left	If heading error is left large, command hard left rudder
Left small	Soft left	If heading error is left small, command soft left rudder
Center	Neutral	If heading error is near zero, command neutral rudder
Right small	Soft right	If heading error is right small, command soft right rudder
Right large	Hard right	If heading error is right large, command hard right rudder

Table 12. Neural-network configurations used in the reported demonstrations.

Model	Config.	Time step (s)	Update rate	Hidden layers	Neurons/layer	Epochs	Batch size
Full-scale	A	0.25	20	2	32	200	64
Full-scale	B	1.00	20	10	50	700	150
L7	A	1.00	5	1	10	50	32
L7	B	1.00	5	1	50	700	150

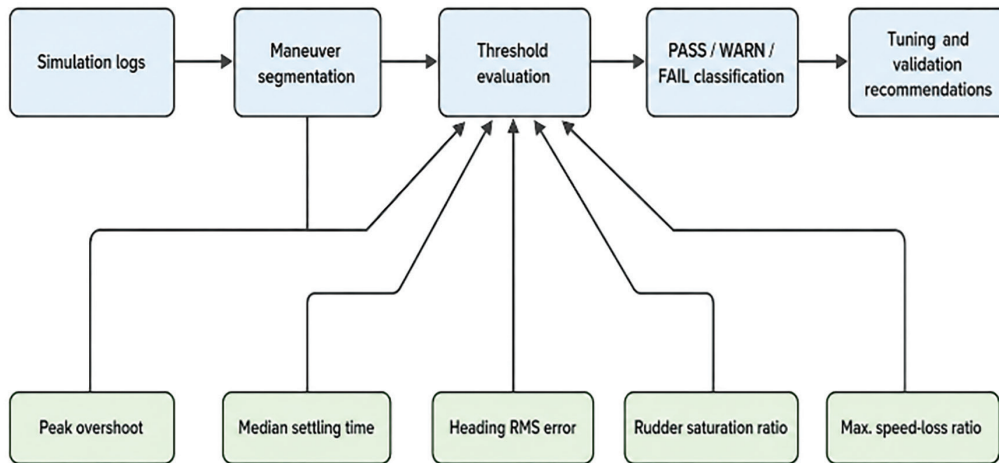


Figure 7. Post-run analyzer concept. Configurable thresholds convert raw simulation traces into engineering-level feedback.

Table 13. Diagnostic metrics used by the automatic post-run analyzer.

Metric	Definition and engineering meaning
Peak overshoot	$M_p = \max_j M_{p,j}$, where each $M_{p,j}$ measures how far the heading exceeds the segment target relative to the initial error magnitude. Large values indicate aggressive and weakly damped turning behavior.
Median settling time	$T_s = \text{median}_j(T_{s,j})$ where $T_{s,j}$ is the first time at which $ e_\psi $ remains inside a tolerance band for a sustained interval. Large values indicate slow recovery after reference changes.
Heading RMS error	$e_{\text{RMS}} = \sqrt{\frac{1}{N} \sum_{k=1}^N e_\psi^2(k)}$. This metric captures the average tracking accuracy over the full simulation run.
Rudder saturation ratio	$R_\delta = 1/N \sum_{k=1}^N 1_{ \delta(k) \geq \eta \delta_{\text{max}} }$, where η is a configurable near-saturation fraction. High values indicate excessive actuator demand.
Maximum speed-loss ratio	$R_u = \max_j \frac{u_{0,j} - u_{\text{min},j}}{u_{0,j} + \varepsilon}$ where $u_{0,j}$ is the baseline segment speed and $u_{\text{min},j}$ the minimum speed reached during the maneuver. High values indicate that heading corrections are degrading forward-motion efficiency.

6.2. Controller-Comparison Results

Figures 12-19 demonstrate that changes in the controller configurations entered through the simulator are reflected in the recorded heading, rudder, and speed responses for both the L7 and full-scale models. Differences in transient behavior and actuator demand are visible among the demonstrated configurations. These cases are intended to demonstrate the controller-testing and analysis workflow rather than to establish the relative performance or superiority of the implemented control methods.

6.3. Full-Scale Simulation on a Selected İstanbul Strait Route Segment

Figures 20-25 demonstrate the complete software workflow using an archived full-scale vessel simulation conducted on a geographically referenced route segment of the İstanbul Strait. The simulated route covers only a limited section of the Strait near a navigationally constrained area and does not represent a complete İstanbul Strait transit.

In this subsection, the term full-scale refers to the vessel dimensions and model parameters rather than the geographic extent of the simulated route. The purpose of this case is to demonstrate the integration of geographic visualization, closed-loop control, signal recording, data inspection, and post-run assessment within the developed software. It is not intended to establish an optimal controller design.

Figure 20 shows the Simulation tab during execution of the selected route. The central map displays the geographically referenced waypoints, the prescribed route, the current vessel position, and the trajectory generated during the maneuver. The surrounding interface panels provide the instantaneous vessel states, actuator values, simulation parameters, and active controller settings. The time-history panels on the right provide live visualization of selected motion and control variables. This arrangement allows the user to observe the spatial vessel motion and the corresponding dynamic response within the same software environment.

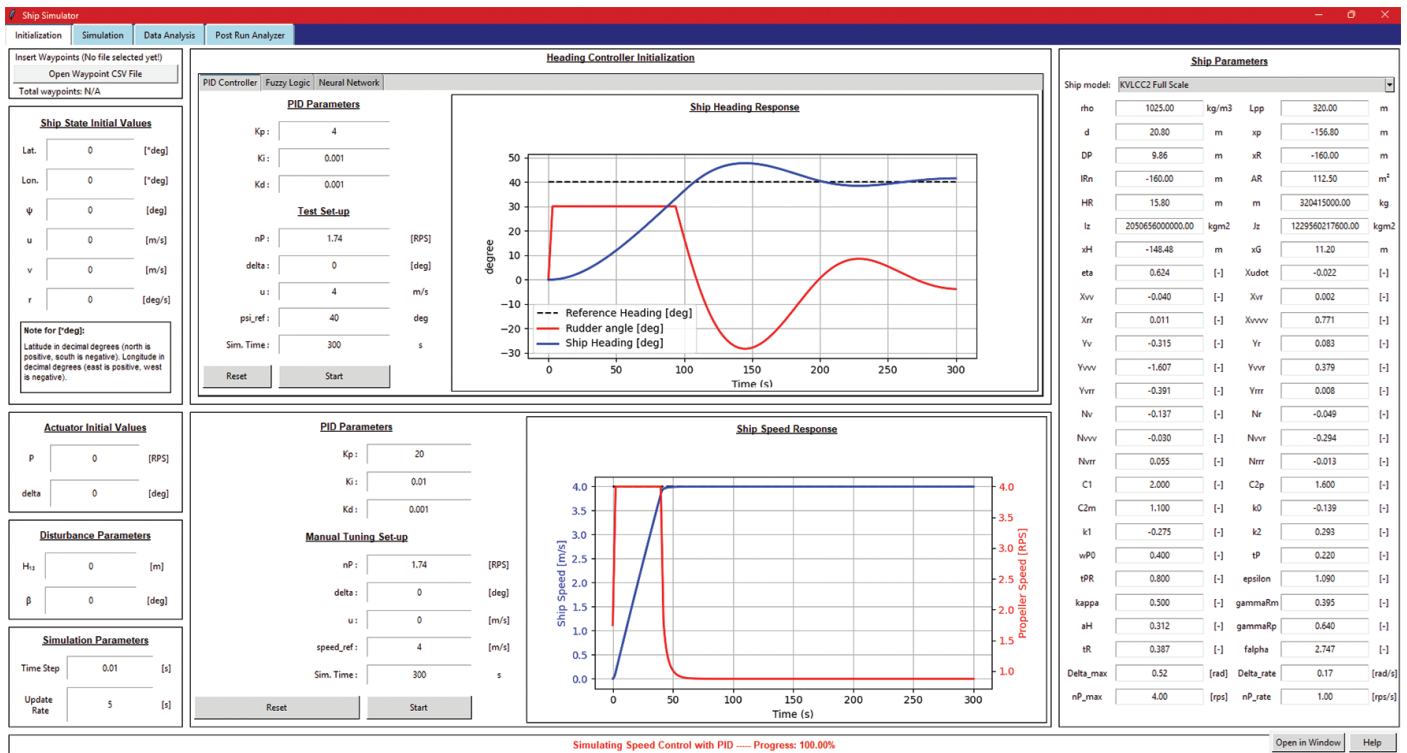


Figure 8. Initialization tab.

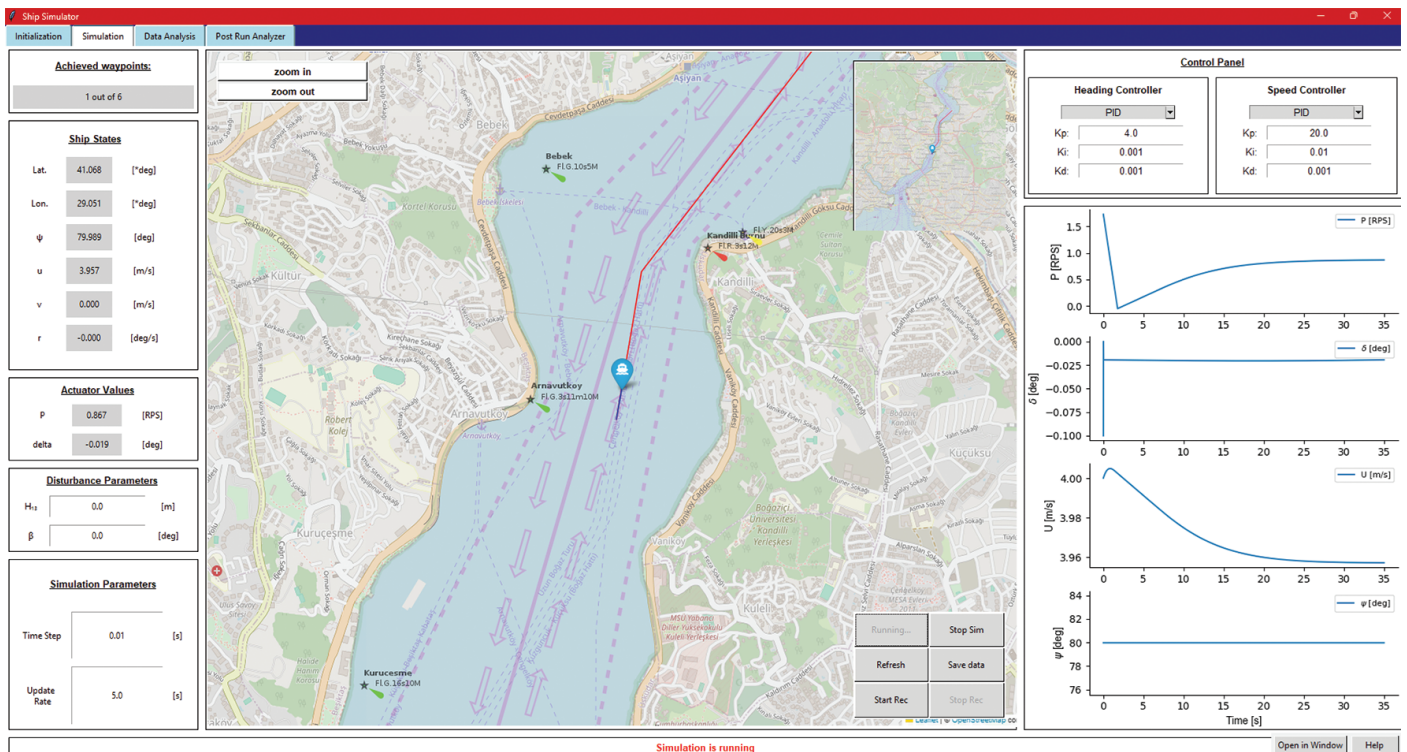


Figure 9. Simulation tab.

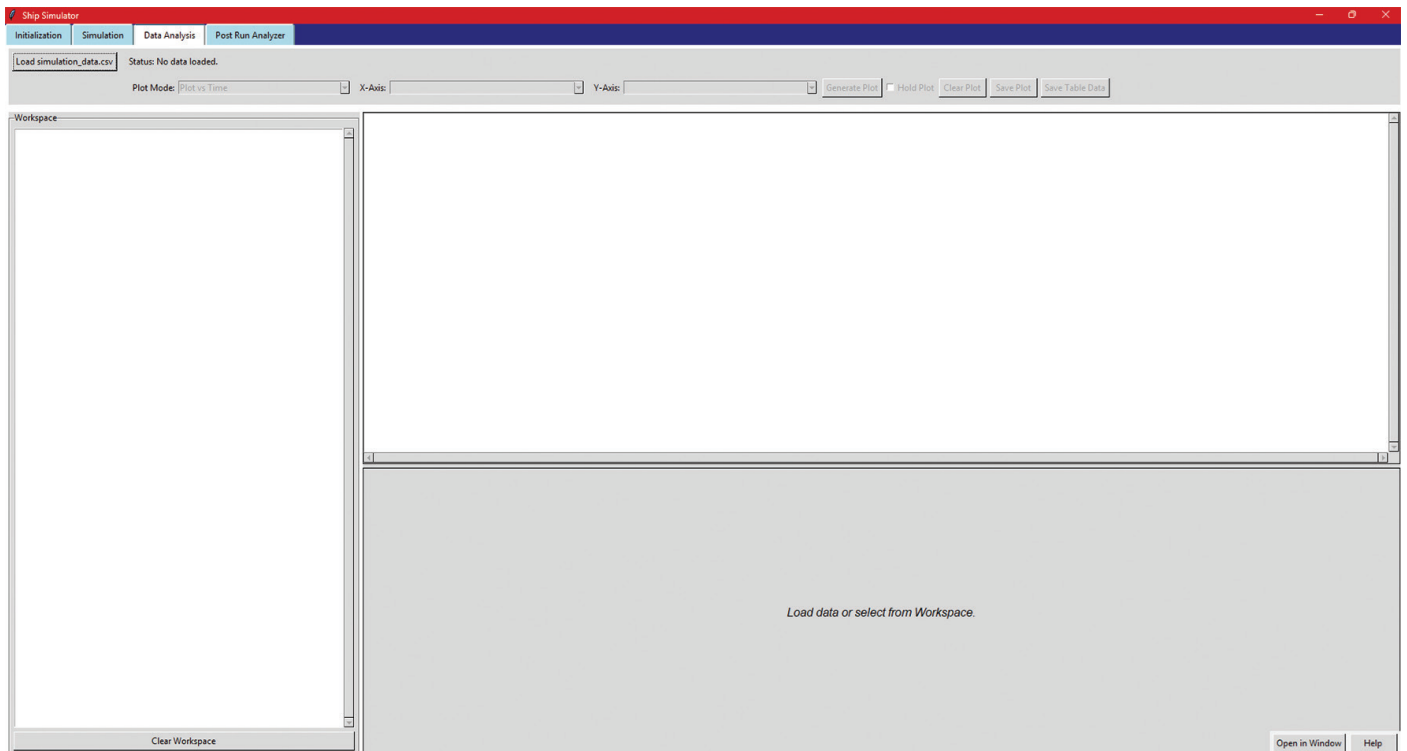


Figure 10. Data analysis tab.

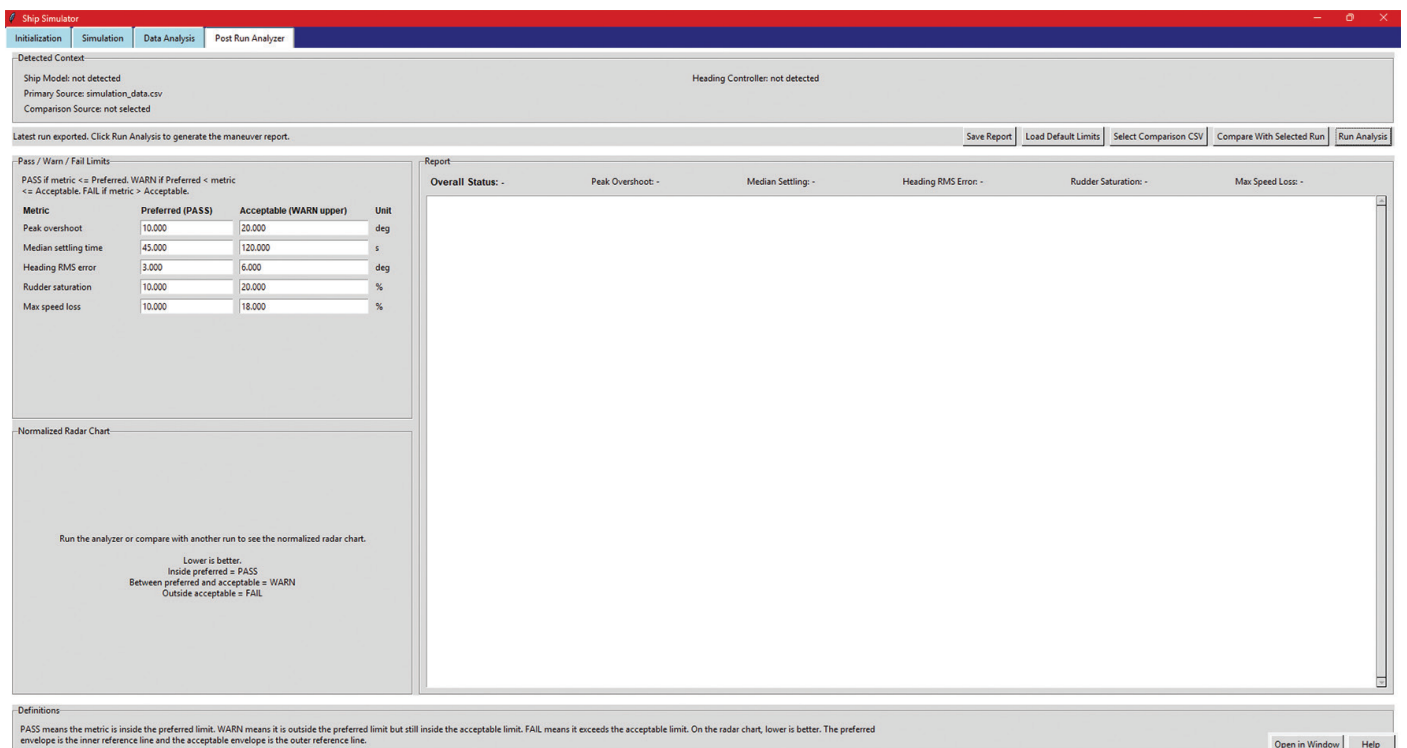


Figure 11. Post-run analyzer tab.

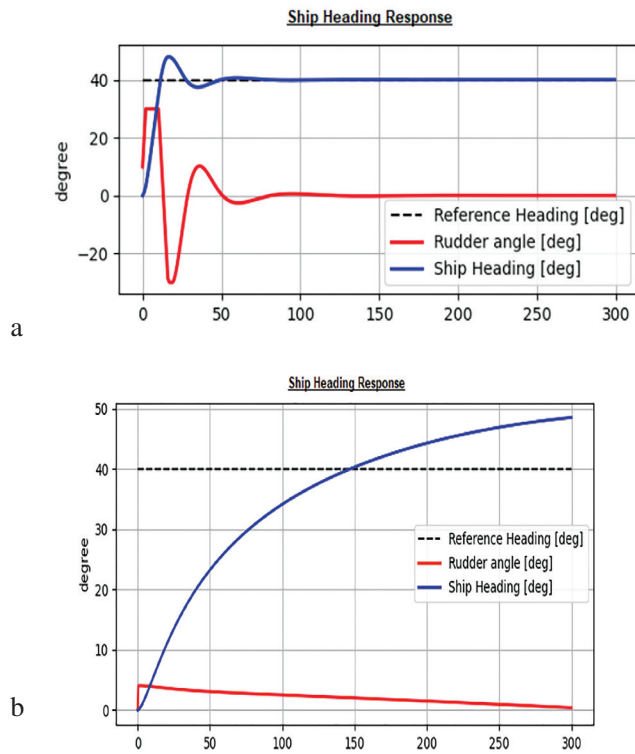


Figure 12. Configuration a) (left) vs. b) (right) for L7-model heading PID tests.

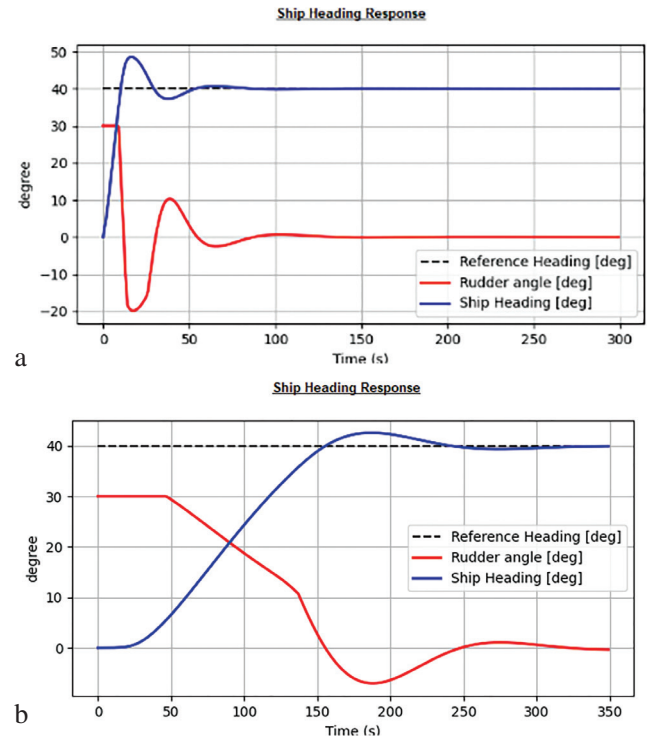


Figure 14. Configuration a) (left) vs. b) (right) for L7-model heading NN tests.

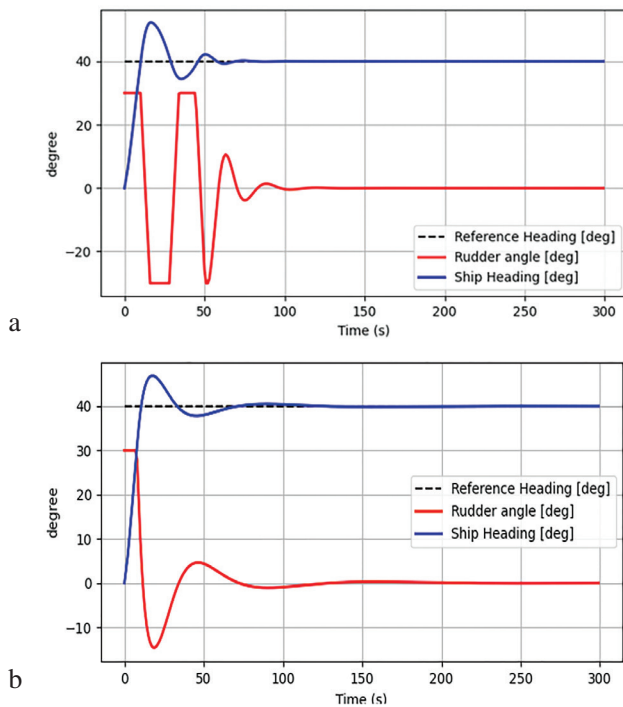


Figure 13. Configuration a) (left) vs. b) (right) for L7-model heading FL tests.

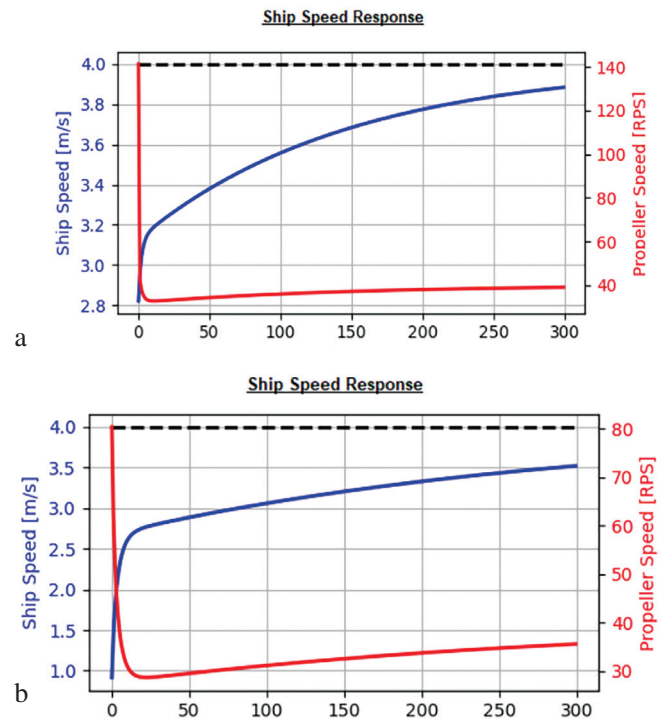


Figure 15. Configuration a) (left) vs. b) (right) for L7-model PID speed tests.

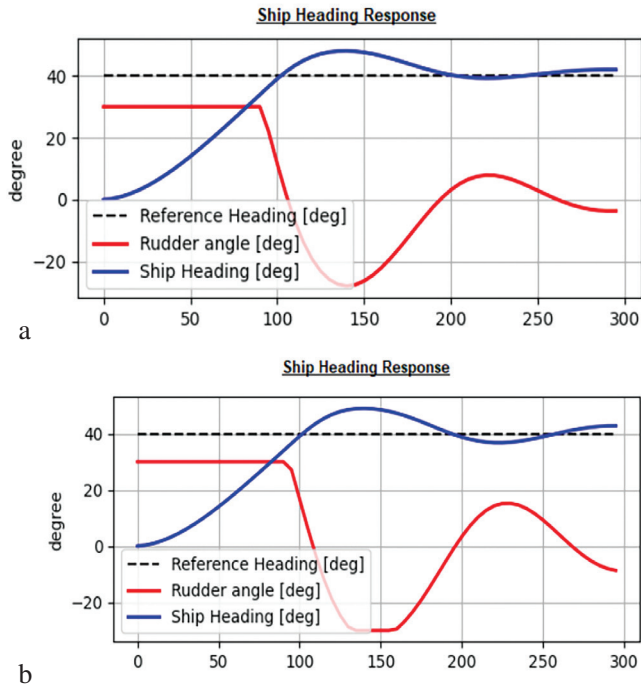


Figure 16. Configuration a) (left) vs. b) (right) for full-scale heading PID tests.

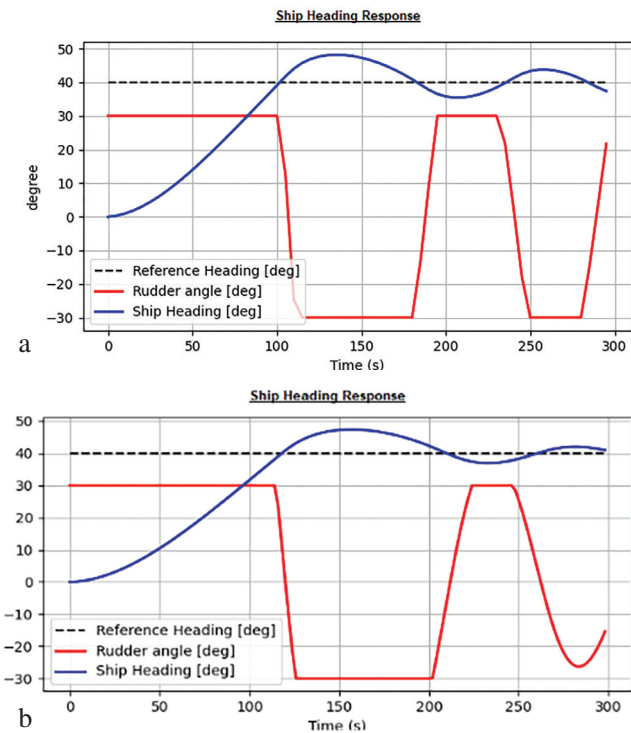


Figure 17. Configuration a) (left) vs. b) (right) for full-scale heading FL tests.

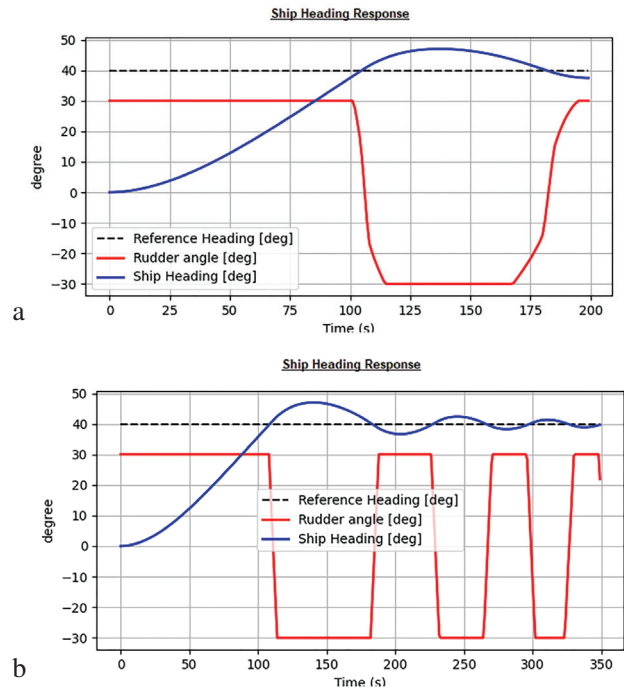


Figure 18. Configuration a) (left) vs. b) (right) for full-scale heading NN tests.

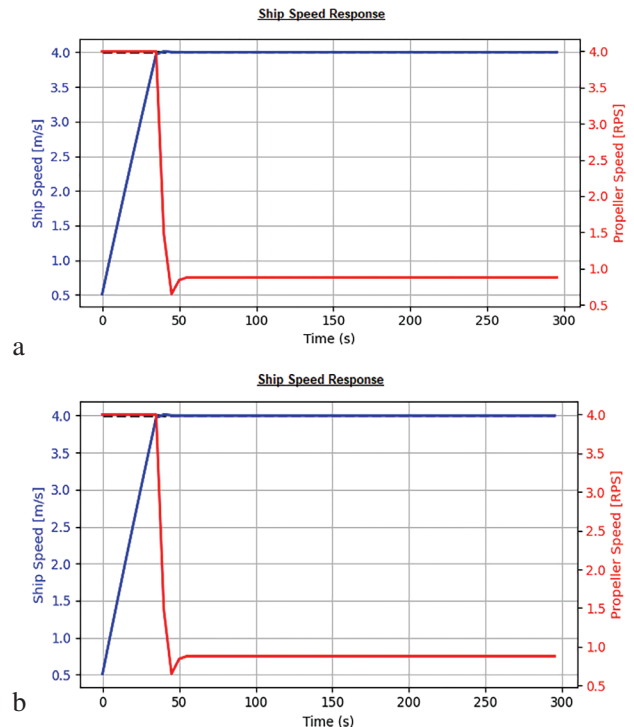


Figure 19. Configuration a) (left) vs. b) (right) for full-scale PID speed tests.

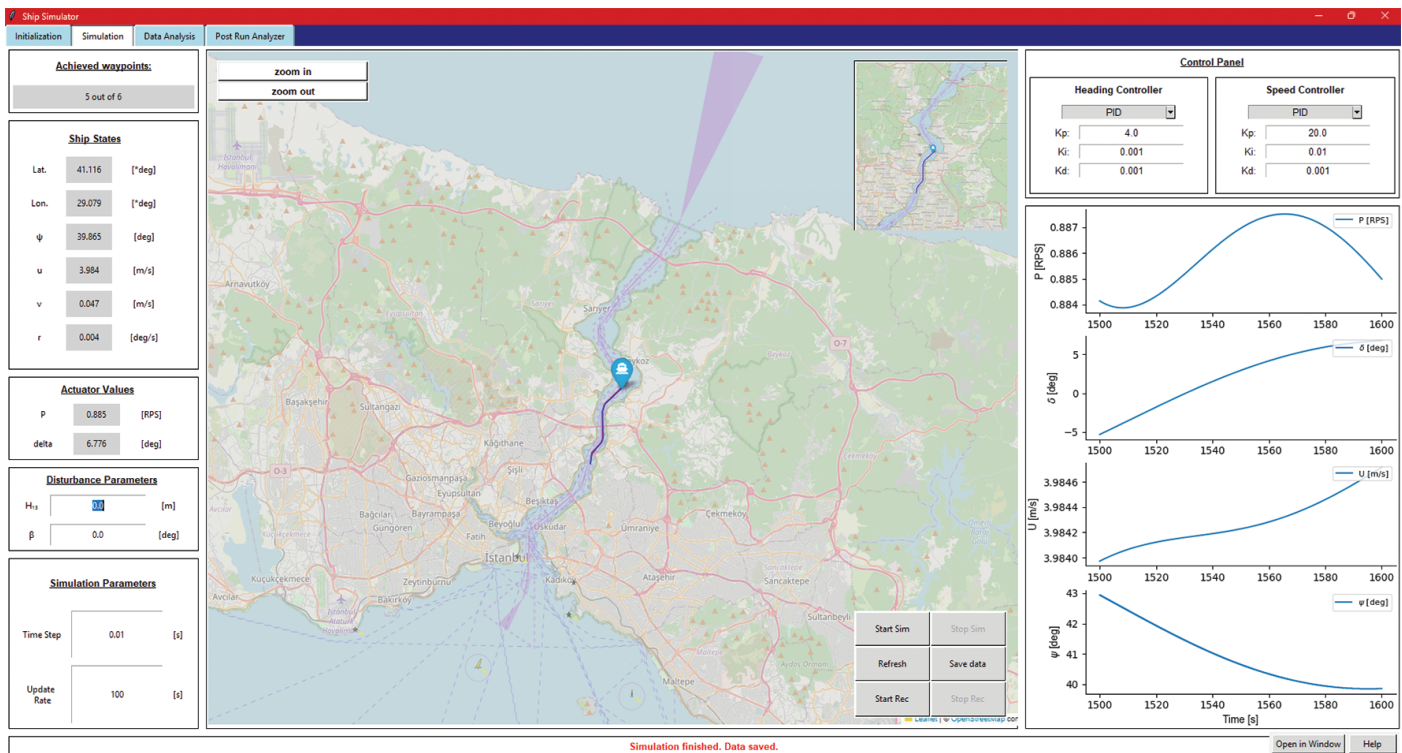


Figure 20. Simulated trajectory along the selected İstanbul Strait route segment in simulation tab.

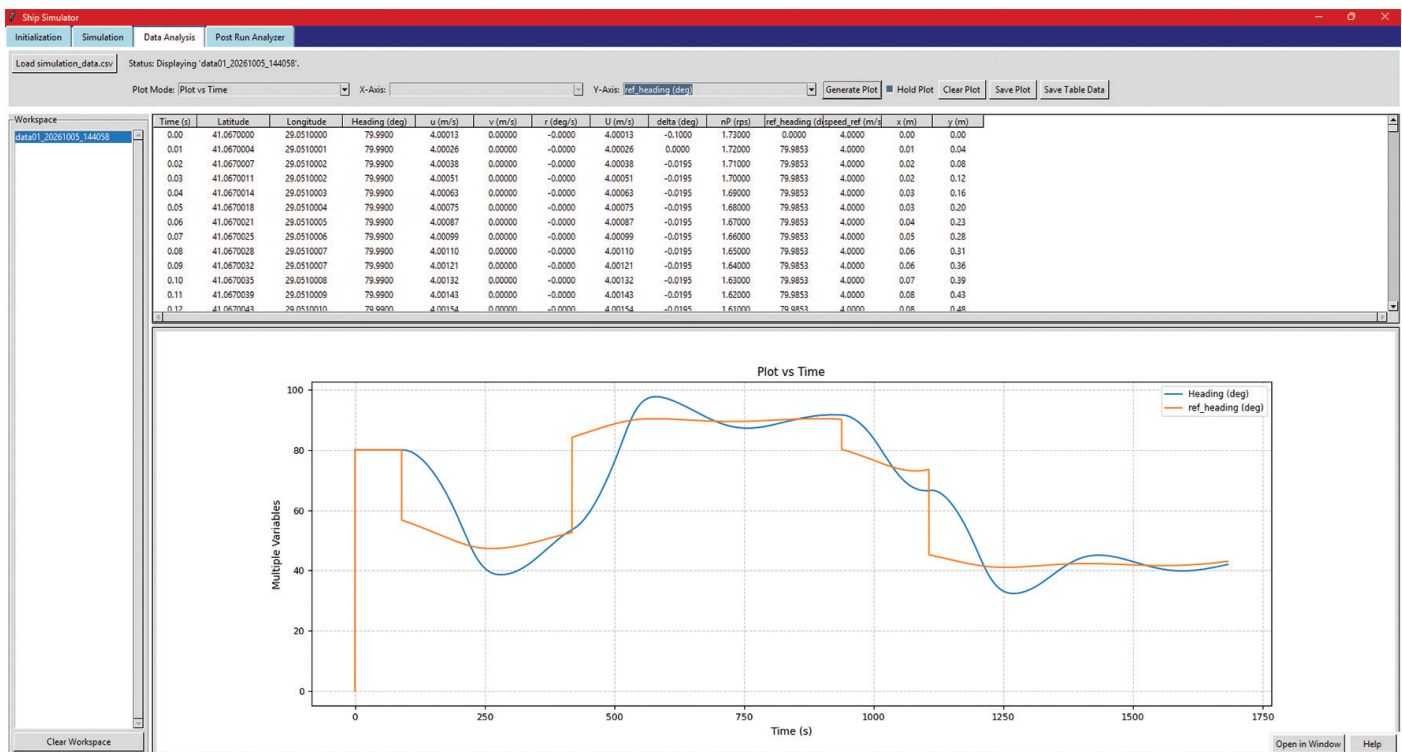


Figure 21. Heading vs. reference in data analysis tab.

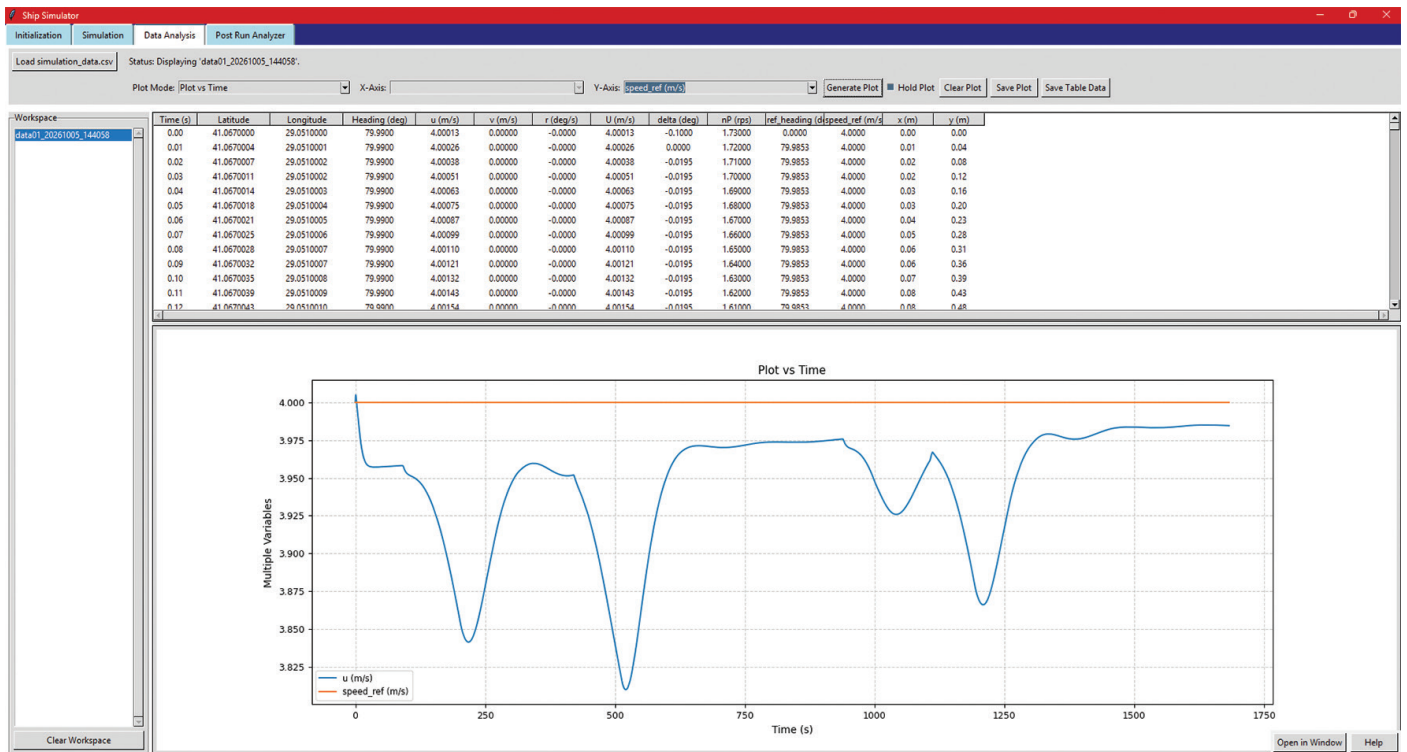


Figure 22. Speed vs. reference in data analysis tab.

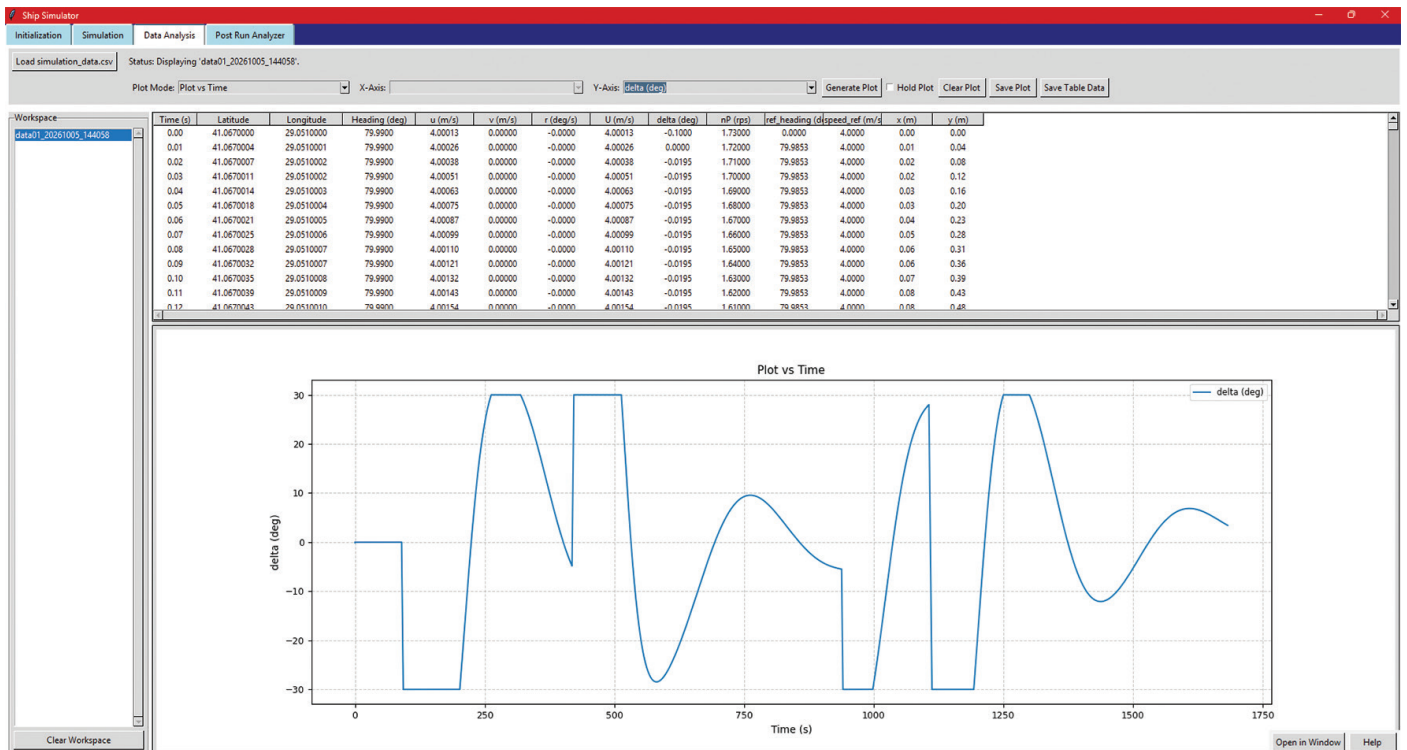


Figure 23. Rudder angle plot in data analysis tab.

Figure 21 presents the recorded vessel heading together with the reference heading generated by the waypoint-guidance module. The reference changes as the active waypoint is updated, producing several distinct turning segments along the route. The vessel follows the general sequence of commanded headings, although overshoot and oscillatory recovery are visible after some reference changes. These results are not presented as evidence of optimized heading control. Instead, they demonstrate that the Data Analysis tab can load an archived simulation, display recorded variables, overlay related signals, and reveal transient characteristics that are not immediately apparent from the geographic trajectory alone.

Figure 22 compares the vessel speed with the commanded reference speed. The measured speed remains relatively close to the reference, with temporary reductions occurring during the more demanding turning sections. The limited speed reduction should not be interpreted as an inconsistency with the heading oscillations shown in Figure 21. Heading and speed are regulated by separate closed-loop controllers. The heading controller generates the rudder command, whereas the speed controller modifies the propeller rotational speed according to the difference between the measured and commanded surge speeds. Their effects remain coupled through the MMG vessel dynamics, but the propulsion controller actively compensates for maneuver-induced changes in resistance and surge speed.

The rudder-angle history in Figure 23 provides the actuator-level information associated with the heading response. Large positive and negative rudder commands occur after several changes in the reference heading, and the rudder reaches its imposed limits during some portions of the maneuver. These periods correspond to the stronger heading corrections visible in Figure 21. The figure therefore demonstrates why route completion alone is insufficient for assessing controller behavior. A vessel may continue along the intended geographic route while requiring substantial steering activity or exhibiting weakly damped heading recovery.

Figure 24 shows that the propulsion input is not held constant during the maneuver. The propeller rotational speed changes throughout the simulation as the speed controller responds to deviations from the reference speed. Increases in propeller command occur around several turning events, indicating active compensation for transient speed loss and additional hydrodynamic resistance. This response explains why the vessel speed in Figure 22 remains comparatively close to its reference despite the pronounced rudder activity and heading oscillations. Figures 21-24 together provide a more complete interpretation of the maneuver by relating heading tracking,

speed regulation, steering demand, and propulsion activity.

Figure 25 presents the same simulation in the Post-Run Analyzer tab. The module processes the recorded signals and reports overall and segment-level indicators, including peak overshoot, median settling time, heading RMS error, rudder-saturation ratio, and maximum speed-loss ratio. The calculated values are compared with user-defined preferred and acceptable limits and are displayed through both a normalized radar chart and a textual diagnostic report. Under the limits configured for this demonstration, the run is classified as FAIL because selected heading-tracking and actuator-use measures exceed the acceptable ranges, even though the speed-loss measure remains comparatively small. This result illustrates the intended function of the post-run module. The simulator does not treat geometric route execution as sufficient evidence of acceptable maneuver quality. Instead, it preserves the complete state and actuator histories and converts them into structured engineering information that can support controller adjustment, comparison between archived runs, and subsequent software-based testing.

6.4. Limitations and Software Availability

The present study has several limitations. The implemented MMG model has not yet been independently evaluated against experimental maneuvering benchmarks, and the PID, FL, and NN configurations were not obtained through a common formal tuning or optimization procedure. Consequently, the demonstrated cases should not be interpreted as validation of hydrodynamic prediction accuracy or as a controlled comparison of controller superiority. The current experiments also do not systematically address environmental disturbances, parameter uncertainty, or robustness across different operating conditions. In addition, the geographically referenced full-scale case is limited to the selected İstanbul Strait route segment described in Section 6.3. These limitations define the scope of the present software demonstration and motivate future benchmark validation, disturbance-rich testing, and controlled controller comparisons. The current research implementation is not publicly distributed as an open-source software package.

7. Conclusion

This paper presented a ship simulator for control testing that combines MMG-based vessel dynamics, waypoint guidance, autonomous and manual control modes, İstanbul Strait-centered visualization, controller comparison, post-processing, and automatic maneuver-quality evaluation. The resulting platform goes beyond a simple route-following demonstrator and functions as a reusable environment for iterative maritime control development.

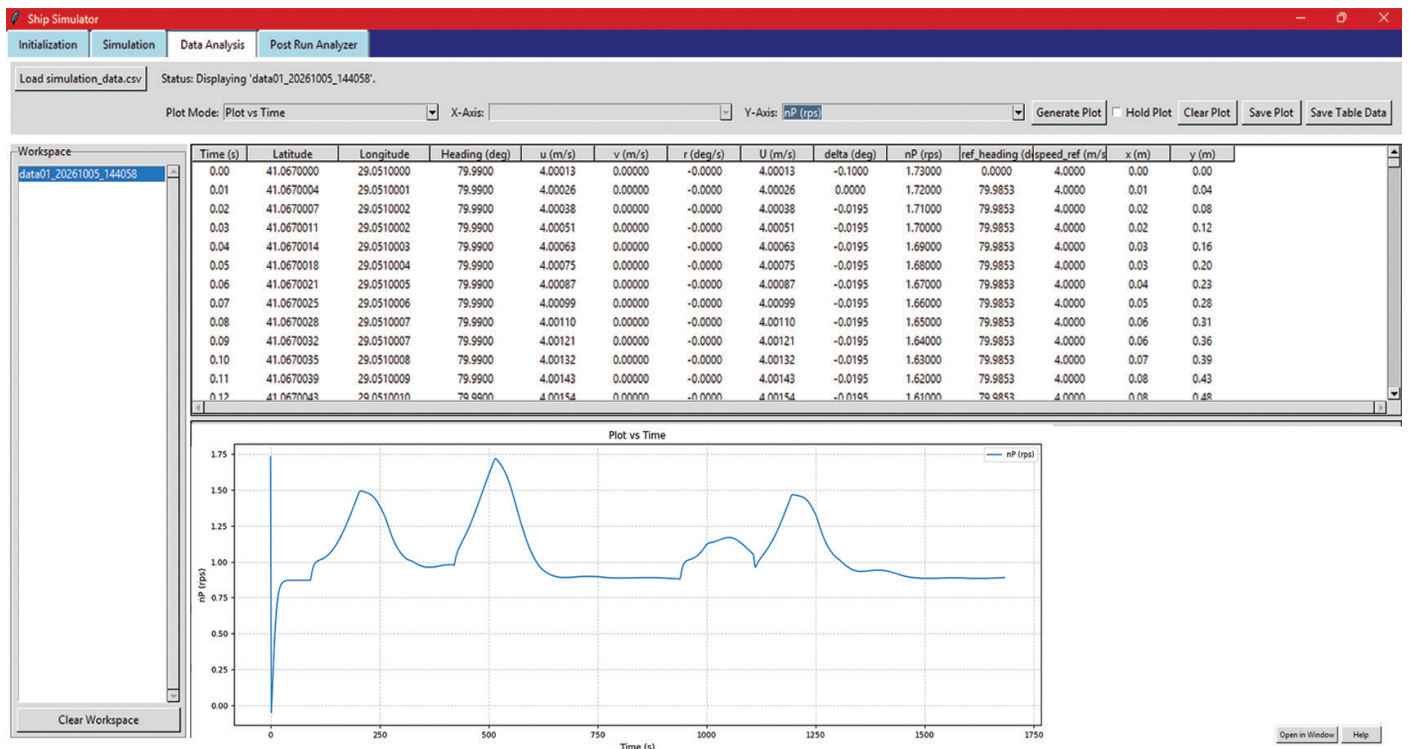


Figure 24. Propeller speed plot in data analysis tab.

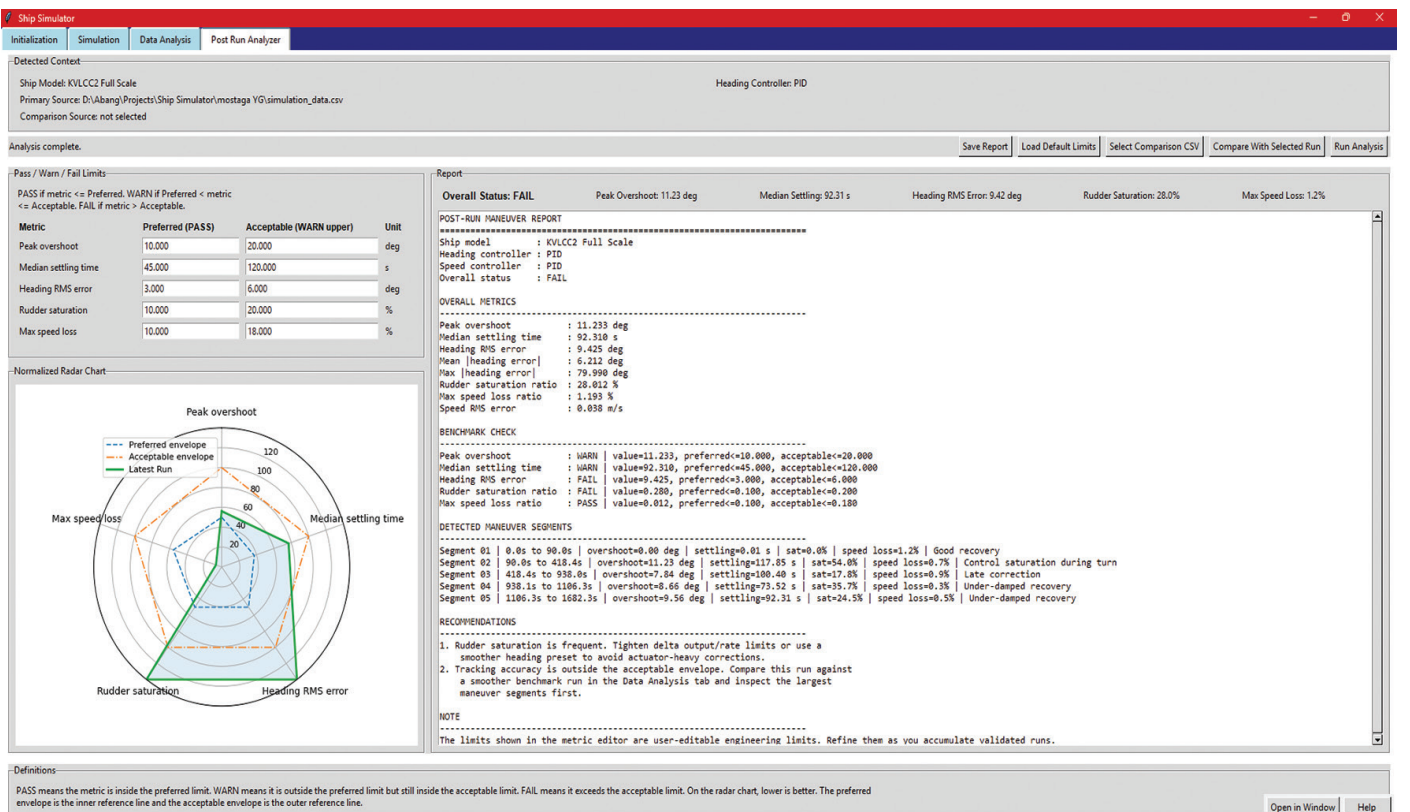


Figure 25. Controller performance result in post-run analyzer tab.

The reported cases demonstrate that the simulator can connect geographically defined routes, MMG-based vessel dynamics, interchangeable controller implementations, synchronized signal recording, and post-run assessment within a common workflow. The PID, FL, and NN examples further show that changes in controller configuration can be examined through the same visualization and analysis structure, although the present demonstrations are not intended to establish controller superiority. The data-analysis and post-run evaluation modules provide quantitative information on tracking response, actuator activity, and speed loss that complements the geographic trajectory. Future work should focus on benchmark validation of the maneuvering model, disturbance-rich scenarios, systematically tuned controller comparisons, and hardware-oriented validation.

Acknowledgements

This study was carried out within the collaboration of the Model-Based Design & Control Systems Laboratory and the Marine Robotics Laboratory at Istanbul Technical University. The work is financially supported by the TÜBİTAK Technology and Innovation Funding Programs Directorate under Program 1507, Project No.: 7240107.

Footnotes

Authorship Contributions

Concept: A. I. Jambak, M. Abdelkhalek, I. Bayezit, and O. K. Kinaci, Design: A. I. Jambak, M. Abdelkhalek, I. Bayezit, and O. K. Kinaci, Data Collection or Processing: A. I. Jambak, M. Abdelkhalek, and I. Bayezit, Analysis or Interpretation: A. I. Jambak, M. Abdelkhalek, and O. K. Kinaci, Literature Review: A. I. Jambak, I. Bayezit, and O. K. Kinaci, Writing: A. I. Jambak, M. Abdelkhalek, I. Bayezit, and O. K. Kinaci.

Conflict of Interest: No conflict of interest was declared by the authors.

Financial Disclosure: The work is financially supported by the TÜBİTAK Technology and Innovation Funding Programs Directorate under Program 1507, Project No.: 7240107.

NOMENCLATURE

Symbol	Description	Unit
m	Ship mass	kg
m_x	Added mass in surge	kg
m_y	Added mass in sway	kg
I_{zG}	Yaw moment of inertia about the center of gravity	kg·m ²
J_z	Added yaw moment of inertia	kg·m ²

x_G	Longitudinal position of the center of gravity	m
u	Surge velocity	m/s
v	Sway velocity	m/s
r	Yaw rate	rad/s
U	Resultant vessel speed	m/s
ψ	Vessel heading	deg
X	Total surge force	N
Y	Total sway force	N
N	Total yaw moment	N·m
X_H	Hull-induced surge force	N
Y_H	Hull-induced sway force	N
N_H	Hull-induced yaw moment	N·m
X_P	Propeller-induced surge force	N
X_R	Rudder-induced surge force	N
Y_R	Rudder-induced sway force	N
N_R	Rudder-induced yaw moment	N·m
ρ	Water density	kg/m ³
L_{pp}	Length between perpendiculars	m
d	Ship draft	m
v'	Non-dimensional sway velocity	–
r'	Non-dimensional yaw rate	–
T	Propeller thrust	N
t_p	Propeller thrust-deduction factor	–
n_p	Propeller rotational speed	rps
D_p	Propeller diameter	m
K_T	Propeller thrust coefficient	–
J_p	Propeller advance ratio	–
w_p	Propeller wake fraction	–
β	Vessel drift angle	rad
δ	Rudder angle	deg
F_N	Rudder normal force	N
t_R	Rudder force-deduction factor	–
a_H	Hull-rudder interaction coefficient	–
x_R	Longitudinal rudder position	m
x_H	Longitudinal position of hull-rudder interaction force	m
A_R	Rudder area	m ²
U_R	Resultant rudder inflow velocity	m/s
u_R	Longitudinal rudder inflow velocity	m/s
v_R	Lateral rudder inflow velocity	m/s
α_R	Rudder inflow angle	rad
γ_R	Rudder flow-straightening coefficient	–

x_{ref}	Reference waypoint x-coordinate	m
y_{ref}	Reference waypoint y-coordinate	m
ψ_{ref}	Reference heading	deg
e_{ψ}	Heading tracking error	deg
u_{ref}	Reference surge speed	m/s
e_u	Surge-speed tracking error	m/s
δ_{cmd}	Commanded rudder angle	deg
$n_{P,cmd}$	Commanded propeller rotational speed	rps
K_p	Proportional controller gain	—
K_i	Integral controller gain	—
K_d	Derivative controller gain	—
M_p	Peak overshoot	—
T_s	Settling time	s
e_{RMS}	Root-mean-square heading error	deg
R_{δ}	Rudder-saturation ratio	—
δ_{max}	Maximum allowable rudder angle	deg
R_u	Maximum speed-loss ratio	—
u_0	Baseline surge speed	m/s
u_{min}	Minimum surge speed during a maneuver	m/s

8. References

- [1] H. Canton, United Nations Conference on Trade and Development (UNCTAD). In The Europa Directory of International Organizations. Springer. 2021.
- [2] M. Stopford, "Maritime economics" (3rd ed.), 2008.
- [3] T. H. Joung, S. G. Kang, J. K. Lee, and J. Ahn, "The IMO initial strategy for reducing greenhouse gas(GHG) emissions, and its follow-up actions towards 2050." *Journal of International Maritime Safety, Environmental Affairs, and Shipping*, vol. 4, no. 1, pp. 1-7, 2020.
- [4] R. Sawada, et al. "Perspective on the marine simulator for autonomous vessel development." arXiv e-prints, 2407.19673. 2024.
- [5] M. Abdelkhalak, et al. "E-VCU software toolbox for ARM cortex-R4 processor based electric vehicle control. XXIX International Conference on Information, Communication and Automation Technologies (ICAT), pp. 1-6, 2023.
- [6] H. Yasukawa, and R. Sakuno, "Application of the MMG method for the prediction of steady sailing condition and course stability of a ship under external disturbances." *Journal of Marine Science and Technology*, vol. 25, pp. 196-220, Mar 2020.
- [7] H. Yasukawa, and Y. Yoshimura, "Introduction of MMG standard method for ship maneuvering predictions." *Journal of Marine Science and Technology*, vol. 20, pp. 37-52, 2015.
- [8] K. Yuen, G. Xu, and J. Lam, "Special issue on 'artificial intelligence & big data in shipping'." *Maritime Policy & Management*, vol. 47, pp. 575-576, 2020.
- [9] M. Maniar, V. Deogaonkar, and A. Somayajula, "Development of a high fidelity multi-agent maritime simulator." *Conference: ASME 2024 43rd International Conference on Ocean, Offshore and Arctic Engineering*, Aug 2024.
- [10] Y. Dai, Y. He, X. Zhao, and K. Xu, "Testing method of autonomous navigation systems for ships based on virtual-reality integration scenarios." *Ocean Engineering*, vol. 309, 118597, 2024.
- [11] M. A. Hinostroza, and A. M. Lekkas, "Temporal mission planning for autonomous ships: design and integration with guidance, navigation and control." *Ocean Engineering*, vol. 297, 117104, Apr 2024.
- [12] H. Xu, and C. Guedes Soares, "Review of path-following control systems for maritime autonomous surface ships." *Journal of Marine Science and Application*, vol. 22, pp. 153-171, Jul 2023.
- [13] A. I. Jambak, M. M. H. Abdelkhalak, I. Bayezit, and O. K. Kinaci, "Development of ship simulator for control testing." 4th International Congress on Ship and Marine Technology (GMO-SHIPMAR 2024), pp. 467-476, 2024.
- [14] A. I. Jambak, and I. Bayezit, "Robust optimal control of a nonlinear surface vessel model with parametric uncertainties." *Brodogradnja: An International Journal of Naval Architecture and Ocean Engineering for Research and Development*, vol. 74, no. 3, pp. 131-143, 2023.
- [15] Y. Jiang, X. Hou, X. Wang, Z. Wang, Z. Yang, and Z. Zou, "Identification modeling and prediction of ship maneuvering motion based on LSTM deep neural network." *Journal of Marine Science and Technology*, vol. 27, pp. 125-137, May 2021.
- [16] I. Bayezit, A. I. Jambak, O. K. Kinaci, and B. Fidan, "Sensor data based system identification and modular autopilot control instrumentation of marine craft." *Ocean Engineering*, vol. 317, 119793, Feb 2025.
- [17] N. Assani, P. Matic, N. Kastelan, and I. Cavka, "A review of artificial neural networks applications in maritime industry." *IEEE Access*, vol. 11, pp. 139823-139848, Jan 2023.
- [18] S. Chang, C. Xiong, and D. Xia, "Algorithm and simulation of ship maneuver scheme in marine environment." In: 2020 15th International Conference on Computer Science & Education (ICCSE), pp. 471-476, 2020.
- [19] T. Mitsuyuki, "MMG 3DOF model identification with uncertainty of observation and hydrodynamic maneuvering coefficients using MCMC method." *Journal of Marine Science and Technology*, vol. 29, pp. 668-682, 2024.
- [20] C. Liu, T. Li, W. Wu, H. Zheng, J. Li, and X. Chu, "Event-triggered predictive path following control of autonomous ships with an MMG model." *Ocean Engineering*, vol. 314, 119582, 2024.
- [21] H. Yu, et al. "Ship path optimization that accounts for geographical traffic characteristics to increase maritime port safety." In: *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, pp. 5765-5776, Jun 2022.
- [22] A. I. Jambak, and I. Bayezit, "Rapid model-based predictive path planning algorithm for autonomous ship in constrained maritime environment." *Ocean Engineering*, vol. 362, 125724, Jul 2026.
- [23] A. B. Ozcan, "Model based navigation and control of a tankship in narrow waterways." 4th International Naval Architecture and Maritime Symposium (INT-NAM), pp. 11-13, 2023.
- [24] A. I. Jambak, I. Bayezit, and M. Reyhanoglu, "Obstacle avoidance for autonomous sailing sea drones with no-go-zone recovery." *Drones*, vol. 10, no. 8, 578, 2026.
- [25] Kongsberg Digital, "Kongsberg Digital," Accessed: Nov 16, 2024. [Online]. Available: <https://www.kongsberg.com>

- [26] Furuno Corporation, "Furuno Envision," Accessed: Nov 16, 2024. [Online]. Available: <https://www.furuno.com/special/en/envision/>
- [27] Furuno Corporation, "Autonomous," Accessed: Nov 16, 2024. [Online]. Available: <https://www.furuno.com/en/marine-dx/autonomous/>
- [28] Furuno Corporation, "Furuno," Accessed: Nov 16, 2024. [Online]. Available: <https://www.furuno.com>
- [29] JRC, "JRC Industry Reports," Accessed: Nov 16, 2024. [Online]. Available: <https://www.jrc.co.jp>
- [30] Wärtsilä Corporation, "Wärtsilä," Accessed: Nov 16, 2024. [Online]. Available: <https://www.wartsila.com>
- [31] Transas Marine, "Transas Marine," Accessed: Nov 16, 2024. [Online]. Available: <https://www.transas.com>
- [32] Siemens AG, "Siemens," Accessed: Nov 16, 2024. [Online]. Available: <https://new.siemens.com>
- [33] Rolls-Royce Marine, "Rolls-Royce," Accessed: Nov 16, 2024. [Online]. Available: <https://www.rolls-royce.com>
- [34] A.I. Jambak, I. Bayezit, O. K. Kinaci, and B. Fidan, "Control-oriented modeling and cascade control of container ships." In: 2023 XXIX International Conference on Information, Communication and Automation Technologies (ICAT), pp. 1-6, 2023.