Doctoral theses at NTNU, 2023:247

Motoyasu Kanazawa

Data-driven enhancement to ship dynamic model

for motion prediction

Doctoral thesis

NTNU
Norwegian University of Science and Technology
Thesis for the Degree of Philosophiae Doctor
Faculty of Engineering
Department of Ocean Operations and Civil Engineering

NTNU
Norwegian University of Science and Technology

Motoyasu Kanazawa

Data-driven enhancement to ship dynamic model

for motion prediction

Thesis for the Degree of Philosophiae Doctor

Ålesund, August 2023

Norwegian University of Science and Technology
Faculty of Engineering
Department of Ocean Operations and Civil Engineering

NTNU
Norwegian University of
Science and Technology

NTNU
Norwegian University of Science and Technology

Thesis for the Degree of Philosophiae Doctor

Faculty of Engineering
Department of Ocean Operations and Civil Engineering

ISBN 978-82-326-7194-6 (printed ver.)
ISBN 978-82-326-7193-9 (electronic ver.)
ISSN 1503-8181 (printed ver.)
ISSN 2703-8084 (online ver.)

IMT-report 2023:247

Doctoral theses at NTNU, 2023:247

Printed by NTNU Grafisk senter

Abstract

Ensuring safety and efficiency is the most critical agenda for the maritime industry. It is essential to have a good understanding of ship dynamics and make accurate predictions of ship motion. This enables us to properly evaluate risks such as collision and excessive motion. The bedrock to do so is to develop advanced ship dynamic models for making accurate motion predictions. Ships are complex systems composed of multi-scale and multi-disciplinary sub-systems, such as a hull, thrusters, engines, etc. They are exposed to external disturbances such as wind, waves, and ocean currents in real-world sea conditions. These complexity and non-linearity lead to strong uncertainties in dynamic models, making it difficult to achieve high performance. The maritime industry has faced increasingly rigorous cost requirements, leading to a demand for a more agile framework for delivering dynamic models. To address this, this dissertation proposes a framework that makes a synergy of two modeling principles.

Physics-based modeling has played a dominant role, emphasizing a mechanical understanding of ship motion. The resulting model is accompanied by physical interpretations and is trusted by experts for their knowledge foundation and reliability. The flip side of its physics-based inspiration is that it tends to perform poorly on the full-scale dataset of interest. Moreover, it requires much expertise and meticulous experiments, resulting in high costs in both time and resources. On the other hand, data-driven modeling aims to learn inductively hidden patterns from full-scale data. It excels at handling strong non-linearity and complexity. With the use of cutting-edge machine learning models that are robust to noise and disturbances, it is possible to effectively utilize onboard data for better ship dynamic models. However, due to their poor interpretability, replacing the strong aspects of physics-based models with them would not be a good idea.

Two modeling principles play different roles and cannot be replaced by each other. Instead, in the agile framework, they need to leverage each other's strengths for complementing each other's weaknesses. To this end, this dissertation develops a framework in which data-driven models enhance the performance of the physics-based model while receiving sufficient knowledge transfer and continuity from the physics-based model. New techniques are presented in four case studies for various pragmatic problems where full-scale data needs to rapidly benefit the performance of the physics-based model. The comprehensive experimental study addresses how the physics-based model and data benefit the enhanced performance in such a framework. A novel method is also presented to bridge the dynamics of similar ships and ease model development. This dissertation proposes specific enablers of such a framework that enhances physics-based models with data-driven methods and demonstrates its promising effectiveness through simulation and real-world validation experiments.

Acknowledgment

The research presented in this thesis was conducted at the Norwegian University of Science and Technology in Ålesund within the Department of Ocean Operations and Civil Engineering (IHB). Financial support was provided by the Research Council of Norway, IKTPLUSS project No.309323 "Remote Control Center for Autonomous Ship Support".

First and foremost, I would like to express my sincere gratitude to my main supervisor Prof. Dr. Houxiang Zhang. He has been kindly offering invaluable guidance, support, and encouragement throughout my PhD journey. His mentorship has motivated me to become an independent researcher completing this work. He has been a true inspiration for me as a researcher. I also would like to convey my appreciation to my co-supervisor Prof. Dr. Guoyuan Li for his insightful feedback and stimulating discussions, which have significantly contributed to the quality of my PhD work. I am thankful to my colleagues and friends for their support throughout my study. Dr. Robert Skulstad and Dr. Tongtong Wang have greatly contributed to my work through everyday discussions. Dr. Peihua Han has always provided feedback with his knowledge. It has been a great friendship with Dr. Baiheng Wu, a love for dogs is something we both have in common. For almost three years, Chunlin Wang has been someone to spend a good time with in Norway. I am fortunate to have Dr. Shuai Yuan, Dr. Luman Zhao, and Dr. Ting Liu, who served as senior figures in my research life. I would also like to extend my thanks to all members of the Intelligent Systems Lab at NTNU Ålesund. I am greatly indebted to Assoc. Prof. Dr. Lars Ivar Hatledal for his support regarding the co-simulation tools. Mingda Zhu, Zizheng Liu, Sihan Gao, Weiwei Tian, Shiyang Li, and Qin Liang, have been comrades in supporting my research journey. I wish to express my gratitude to Kana Banno and Jingwen Ding, who, based on their experience in Norway, provided assistance in my life here. My heartfelt gratitude also goes to Assoc. Prof. Dr. Ryota Wada and Dr. Yasuo Ichinose, who have been a constant source of support for me personally and professionally. I am deeply grateful to Prof. Dr. Jianwei Zhang for warmly welcoming me into the Technical Aspects of Multimodal Systems (TAMS) group at University of Hamburg. Gaining new insights into control and data-driven models in Hamburg was crucial in completing this dissertation. My thanks go out to everyone in the TAMS group. The support from communities in Japan has been a significant encouragement for me. I am grateful to my longtime Japanese friends who have consistently provided me with new inspirations. I would like to extend my thanks to the Nippon Foundation, which supported my endeavors abroad during my master's program, providing an essential foundation for my overseas experience.

Finally, I would like to express my heartfelt appreciation to my family for their priceless love. Their unwavering support has been a deriving force of my motivation and inspiration.

Contents

Abstract i

Acknowledgment iii

List of Publications ix

List of Abbreviations xi

Nomenclature xiii

List of Figures xv

List of Tables xix

1 Introduction 1
- 1.1 Background and motivation . . . 2
  - 1.1.1 Physics-based models as knowledge foundation . . . 2
  - 1.1.2 Data-driven models . . . 3
  - 1.1.3 Data-driven enhancement to foundation models . . . 4
  - 1.1.4 Research questions . . . 6
- 1.2 Scope of work . . . 8
  - 1.2.1 Research objectives . . . 8
  - 1.2.2 Interconnection between the research objectives . . . 9
- 1.3 Contributions of the dissertation . . . 9
- 1.4 Structure of the dissertation . . . 10

2 Data-driven enhancement to the physics-based model 13
- 2.1 Data-driven enhancement to the foundation model . . . 14
  - 2.1.1 Related works in interdisciplinary fields . . . 14
  - 2.1.2 Four modes of enabling the DE framework . . . 16
  - 2.1.3 One-shot or iterative . . . 19
- 2.2 The principles of physics-based models . . . 20
  - 2.2.1 Maneuvering model . . . 20
  - 2.2.2 Stochastic model for linear ship response . . . 21
- 2.3 The principles of data-driven models . . . 22

2.3.1 Data-driven models . . . 22
2.3.2 Hyperparameter optimization . . . 24
2.4 Experimental platforms . . . 25
2.4.1 The R/V Gunnerus . . . 25
2.4.2 Co-simulation . . . 27
2.4.3 MSS toolbox . . . 28
2.5 Chapter summary . . . 28

3 Case study: Enhancement to full-scale operations 29
3.1 One-shot enhancement with consideration for command assumption . . . 29
3.1.1 Methodology . . . 30
3.1.2 Validation experiment . . . 31
3.2 Enhancement to full-scale docking operations . . . 34
3.2.1 Methodology . . . 35
3.2.2 Validation experiment . . . 36
3.3 Pre-training method for data-driven enhancement . . . 40
3.3.1 Methodology . . . 41
3.3.2 Validation experiment . . . 43
3.4 Enhancement to the full-scale DP operation . . . 46
3.4.1 Methodology . . . 47
3.4.2 Validation experiment . . . 50
3.5 Chapter summary . . . 52

4 Case study: Investigating the role of foundation models and data in enhanced performance 55
4.1 Experimental investigation . . . 55
4.1.1 Cooperative ship model . . . 55
4.1.2 Simulation experiment . . . 59
4.1.3 Full-scale experiment . . . 64
4.2 Chapter summary . . . 66

5 Case study: Knowledge transfer between similar ships 69
5.1 Knowledge transfer between similar ships . . . 70
5.1.1 Methodology . . . 70
5.1.2 Model refinement . . . 72
5.1.3 Validation experiment . . . 74
5.1.4 Discussion . . . 78
5.2 Chapter summary . . . 80

6 Conclusion and further work 81
6.1 Summary of contributions . . . 81

6.2 Summary of publications 82
6.3 Future work 83

References **85**

Appendix **93**

A Paper I **95**

B Paper II **109**

C Paper III **127**

D Paper IV **139**

E Paper V **153**

F Paper VI **165**

List of Publications

This thesis is based on research resulting in four journal papers and two conference paper. They are all enclosed in the appendix section. In the following list of publications, the papers are listed chronologically, but in the main body of the text a more thematic presentation order is prioritized over the chronological one.

I M. Kanazawa, R. Skulstad, G.Li, L. Hatledal, and H. Zhang, "A multiple-output hybrid ship trajectory predictor with consideration for future command assumption", *IEEE Sensors Journal*, vol.21, issue.23, pp. 27124–27135, 2021.

II M. Kanazawa, L. Hatledal, G. Li, and H. Zhang, "Co-simulation-based Pre-training of a Ship Trajectory Predictor", *Software Engineering and Formal Methods. SEFM 2021 Collocated Workshops. Conference proceedings*, pp. 173–188, September 2022.

III M. Kanazawa, R. Skulstad, T. Wang, G. Li, L. Hatledal, and H. Zhang, "A Physics-Data Co-Operative Ship Dynamic Model for a Docking Operation", *IEEE Sensors Journal*, vol.22, issue.11, pp. 11173–11183, 2022

IV M. Kanazawa, T. Wang, R. Skulstad, G. Li, and H. Zhang, "Knowledge and data in cooperative modeling: Case studies on ship trajectory prediction", *Ocean Engineering*, vol.266, part.4, 2022

V M. Kanazawa, T. Wang, Y. Ichinose, R. Skulstad, G.Li, and H. Zhang, "Bridging similar ships' dynamics for safeguarding the system identification of maneuvering models", *Ocean Engineering*, vol.280, 2023

VI M. Kanazawa, T. Wang, R. Skulstad, G. Li, and H. Zhang, "Physics-data cooperative ship motion prediction with onboard wave radar for safe operations", *the 32nd International Symposium on Industrial Electronics (ISIE)*, Accepted, 2023.

The following papers will not be discussed in this thesis. They may, however, be considered relevant due to co-authorship and similar topics:

i T. Wang, L. Hatledal, M. Kanazawa, G. Li, and H. Zhang, "Effect of Ship Propulsion Retrofit on Maneuverability Research Based on Co-simulation", *Springer Lecture Notes in Computer Science*

ii T. Wang, R. Skulstad, M. Kanazawa, G. Li, V. Æsøy, and H. Zhang, "Physics-informed Data-driven Approach for Ship Docking Prediction", *IEEE International Conference on Real-time Computing and Robotics (RCAR)*, July 2022.

iii T. Wang, R. Skulstad, M. Kanazawa, L. Hatledal, G.Li, and H. Zhang, "Adaptive Data-driven Predictor of Ship Maneuvering Motion Under Varying Ocean Environments", *Leveraging Applications of Formal Methods, Verification and Validation Practice 11th International Symposium (ISoLA)*, Rhodes, Greece, October 2022.

iv T. Wang, R. Skulstad, M. Kanazawa, G. Li and H. Zhang, "Knowledge Transfer Strategy for Enhancement of Ship Maneuvering Model", Accepted, *Ocean Engineering*, 2023.

List of Abbreviations

DE	Data-driven enhancement to the foundation model
DP	Dynamic positioning
DoF	Degree of freedom
DT	Decision tree
ECS	Entity-component system
FMI	Functional mock-up interface
FMU	Functional mock-up unit
GNC	Guidance, navigation and control
GNSS	Global navigation satelite system
IMO	The international maritime organization
ITTC	The international towing tank conference
KF	Kalman filter
LSTM	Long short-term memory
MLP	Multilayer perceptron
MMG	Mathematical modelling group
MPC	Model predictive control
MRU	Motion reference unit
MSS	Marine systems simulator
MSE	Mean squared error
ML	Machine learning
NN	Neural network
OSP	Open simulation platform
RAO	Response amplitude operator
RO	Research objective

RNN	Recurrent neural network
RQ	Research question
R/V	Research vessel
SA	Situation awareness
SVM	Support vector machine
TPE	Tree-structured parzen estimator

Nomenclature

η	NED position and heading vector
ν	Linear and rotational velocity vector
ψ	Heading angle
R	Rotation matrix
M	Mass matrix
C	Coriolis-centripetal matrix
D	Damping matrix
τ	Force vector
ρ_a	Density of air
$C_{X/Y/N}$	Wind force coefficient matrices
V_w, V_{rw}	Wind velocity, velocity relative to the ship
β_w	Wind angle
γ_w	Wind angle relative to the ship bow
A_{FW}	Projected longitudinal area of the ship above water
A_{LW}	Projected lateral area of the ship above water
L_{oa}	Overall length of the ship
$a()$	Activation function
w	Weight vector
y	Model output
f	Function of the foundation model
g	Function of the data-driven model
$\mathcal{L}$	Modified loss function
$\mathcal{L}_t$	Loss function for the discrepancy between the output and target vector.

$\mathcal{L}_s$	Loss function for the discrepancy between the output vectors from the data-driven model and the physics-based model on the simulator.
λ	A hyperparameter that balances $\mathcal{L}_t$ and $\mathcal{L}_s$
$\mathcal{D}_a$	Dataset with augmented and real-world data
$\mathcal{D}$	Dataset with real-world data
$\mathcal{D}_s$	Dataset with augmented data
θ	A set of parameters of the foundation model
$\Delta\theta$	An update to θ
S_i	The energy spectrum of ship motion
S_w	The directional wave spectrum
θ_w	The relative direction of waves
m_i	The zeroth-order moment of the ship motion spectrum
$\hat{\sigma}_i^m$	The estimation of the standard deviation of ship response
h_t	Hidden state vector at time t in LSTM
c_t	Cell state vector at time t in LSTM
i_t, f_t, g_t, o_t	The input, forget, cell and output gates at time t in LSTM
σ	Sigmoid function

List of Figures

1.1 Illustration of ship motion prediction by the physics-based foundation model and data-driven model. 1
1.2 A schematic overview of the traditional framework for identifying ship dynamic models for different tasks. 3
1.3 A schematic overview of the proposed framework. 4
1.4 Overview of the interconnections between the research objectives and publications. Papers notated with the capital Roman number indicate publications that this dissertation is mainly based on. Those with the lower Roman number indicate associated publications the author contributed as a co-author. 11

2.1 Illustration of the data-driven enhancement of the foundation model. . . 13
2.2 The update mode. 16
2.3 The convert mode. 17
2.4 The serial mode. 18
2.5 The parallel mode. 18
2.6 An example structure of MLPs. 23
2.7 Illustration of an LSTM cell. 24
2.8 A snapshot of the starboard view of the R/V Gunnerus. 25
2.9 A diagram of FMUs and their connections for the entire R/V Gunnerus on Vico. 27

3.1 A schematic structure of the proposed ship motion predictor with consideration for future command assumption. 30
3.2 Time series of thruster revolution and angle for validation study. 31
3.3 Time series of prediction errors S_{ij} made by (orange) the proposed hybrid predictor, (gray) the foundation model, and (blue) the pure data-driven predictor without using the foundation model. 33
3.4 Snapshots of the predicted and true trajectories at (A) and (B) shown in the time series in Fig. 3.3. 33
3.5 Mean and 90% boundary value of prediction error over the 30s prediction horizon in the test dataset. 34
3.6 A schematic overview of the data-driven enhancement to the foundation model for docking operations. 35

3.7 Results of ablation studies of input features of supportive data-driven models. 37
3.8 Time histories of the surge, sway, and yaw velocities of the example operation in the test dataset. 38
3.9 Time histories of prediction errors of the example operation in the test dataset at 30s future made by the foundation model and the enhanced model. 38
3.10 30s true and predicted trajectories at time step (A), (B), and (C) of the example operation in the test dataset shown in Fig. 3.9. 39
3.11 Mean and 90% percentile of prediction errors over the 30s prediction horizon. 40
3.12 The framework of co-simulation-based development of a ship trajectory predictor. 41
3.13 The diagram of the definition of the North-East-Down frame and the body-fixed frame when making prediction. 41
3.14 Pre-defined time series of thruster angle given by a zig-zag controller. . . 43
3.15 Mean prediction errors in the test dataset. 45
3.16 (left) Snapshots of 30s prediction at $t = 0$s, $t = 30$s, and $t = 60s$ of one scenario in the test dataset (right) Time histories of vessel state and commands. 46
3.17 A schematic overview of the proposed approach. 47
3.18 Sub datasets for the case study. 49
3.19 True versus predicted values in the (a) heave, (b) pitch, and (c) roll directions in the test datasets for all case studies. (green: Physics) predictions only with the physics-based model. (red: Physics-data) predictions with the physics-based model and its data-driven mapping by using the original training datasets. (blue: Physics-data-US) predictions with the physics-based model and its data-driven mapping by using the UnderSampled (US) training datasets. 50

4.1 An overview of the cooperative model. Sensor data (current ship's states, thruster command values, and wind information) are given to the physics-based model and data-driven compensator. The physics-based model is a 3DOF maneuvering model outputting a trajectory prediction. The data-driven compensator, of which input-output relationship is shown in (4.3), compensates multiple-step-ahead position errors made by the physics-based model. 56
4.2 A schematic relationship between the body-fixed coordinate when making a prediction and the inertial coordinate. The data-driven compensator compensates for position errors made by the physics-based model in the body-fixed coordinate when making a prediction. 56
4.3 Experimental setting. 59

4.4 Snapshots of trajectory predictions made by (a) the pure data-driven model, (b) the cooperative model with one of the moderately-uncertain physics-based models, and (c) the cooperative model with one of the highly uncertain physics-based models with sub dataset $\mathcal{D}_{10}$, $\mathcal{D}_{30}$, and $\mathcal{D}_{100}$. 61
4.5 The effect of the errors of the physics-based model A and the data amount N_D on the errors made by the cooperative model H in the test dataset. . 62
4.6 A projected 2D graph of the effect of the accuracy of the physics-based model A and the data amount N_D on the errors made by the cooperative model H in the test dataset. 63
4.7 Snapshots of trajectory predictions at (a) 20s, (b) 40s, and (c) 60s of the maneuver in the test dataset. 65
4.8 A time series of thruster angle δ and revolution n in the example maneuvers in the test dataset shown in Fig. 4.7. 65
4.9 The prediction error H over the prediction horizon made by pure data-driven, pure physics-based, and cooperative models. 66

5.1 RMSEs in the single-step-ahead velocity prediction in the surge, sway, and yaw directions made by the present MR, SP, and four baseline algorithms. Results here are for the low-speed case study with one of the five training datasets. Note that five lines with five different training datasets for MR highly overlap. 77

List of Tables

2.1 Advantages and disadvantages of physics-based and data-driven modeling principles. 15
2.2 Basic specifications of the R/V Gunnerus for the short and elongated versions. 26
2.3 Onboard data channels of the R/V Gunnerus. 26
2.4 Methodologies and experimental platforms in publications in this dissertation. 28

3.1 A summary of prediction errors in the distance in the case study. 39

5.1 Specifications of two ships. 74
5.2 A comparison of performances of the present model refinement (MR) and five baseline algorithms for two turning circle maneuvers in the test dataset. Steady Turning Radius (STR), Maximum Transfer (MT), Maximum Advance (MA), and Velocity Loss on Steady turn (VLS) defined by the ITTC are compared. The average absolute error is that made by five different training datasets. The standard deviation of absolute error is its standard deviation. N/A indicates one or more trajectories made by five training datasets result in physically unreasonable trajectories. The red numbers show the best performance against the other frameworks for each metric of each maneuver. 76

1 Introduction

The safety and efficiency during ship operations have been the primary agenda in the maritime industry [1]. Situation awareness (SA) is the practice to evaluate "what will happen in the future" and make use of it for better decision making to ensure such operations. The bedrock in obtaining helpful SA is to make an accurate ship motion prediction [2] with a better understanding of ship dynamics, which is the focus of this dissertation.

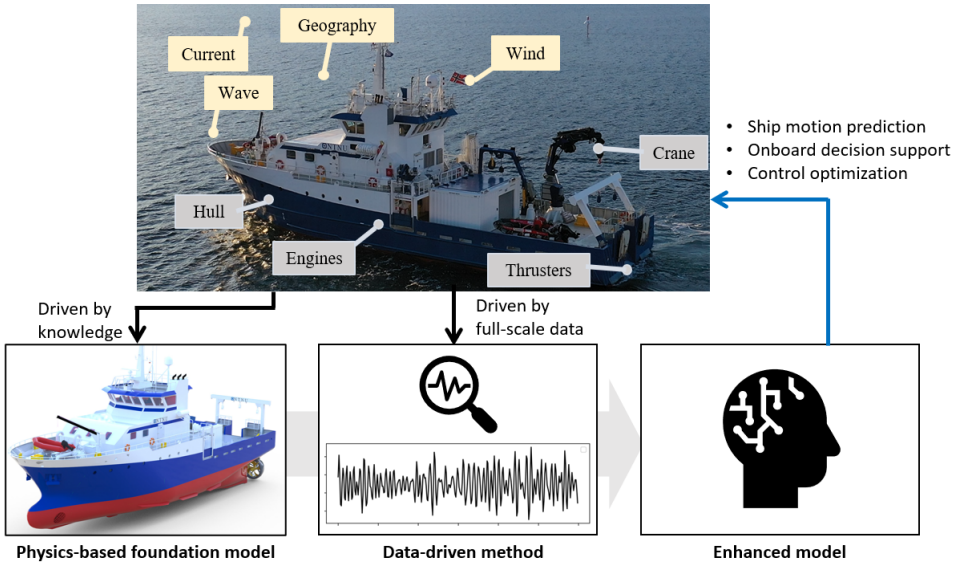


Figure 1.1: Illustration of ship motion prediction by the physics-based foundation model and data-driven model.

For more accurate motion prediction, building ship dynamic models has been a key topic. As shown in the top panel in Fig. 1.1, however, it is not a straightforward task. Ships are comprehensive systems that involve multiple subsystems, such as a hull, engines, thrusters, cranes, and so on. They are multi-scale and multi-disciplinary subsystems communicating with each other. Fortunately, we can model ship dynamics through knowledge-driven approaches that emphasize our expertise, project experiences, and model-scale experiments. Such a modeling practice is known as physics-based modeling strongly derived from mechanistic principles, offering instructive insights acting as

a knowledge foundation. Since our knowledge of the interaction between subsystems and complex environment is limited, it always comes with a challenge in achieving a good accuracy of the entire-ship model. At the same time, we should not overlook the fact that modern ships are equipped with various onboard sensors [3], such as the global navigation satellite system (GNSS), motion sensors, anemometers, and so on. Thanks to the recent advancements in machine learning (ML), they become powerful tools that provide us with valuable feedback from the real world, being rapidly applied to the maritime domain [4] recently. These two genres of modeling are based on distinguished principles, namely physics-based and data-driven. The former is where we place our understanding and reliability, while the latter respects inductive learning for better performance. Both elements are necessary for representing comprehensive and complex systems in the real world. However, if one tries to replace the other's role, it becomes costly and time-consuming.

In this dissertation, instead of that, we explore the collaboration between them who play their respective roles for maximizing their own values. Assuming specific physics-based models are available acting as a knowledge foundation, the author's core idea is to enhance its performance with state-of-the-art data-driven models powered by onboard sensor measurements. This dissertation is tasked with proposing and investigating enablers to do so for reconciling the data-driven insights with such a knowledge foundation.

1.1 Background and motivation

In this dissertation, ship motion prediction refers to forecasting the ship's future states based on an understanding of its dynamics. The international maritime organization (IMO) requires compliance with Standards for Ship Maneuverability for ships built after 2004 [5], highlighting the importance of understanding ship dynamics and predicting ship motion for ensuring safety at sea. More specifically, ship motion prediction is one of the pillars for assessing collision risk with marine obstacles during maneuvering [2]. Ships may lose their position information due to GNSS malfunction. In such cases, predicting their position by using models is known as dead-reckoning tasks [6]. For optimizing ship control commands, the model predictive control (MPC) has also gained attention utilizing ship motion prediction [7,8]. Ship motion prediction is also important for ensuring safety during Dynamic Positioning (DP) operations, where ships maintain their position and heading by using actuators. Accurately predicting ship motion under waves enables emergency actions when large motion is expected [9]. Hence, ship dynamic models for motion prediction are essential for ensuring operational efficiency and safety. This section describes two principles in modeling, leading to the motivation of this work using the data-driven model for enhancing the physics-based model acting as a knowledge foundation.

1.1.1 Physics-based models as knowledge foundation

A physics-based model is a parametric model inspired by mechanistic principles. It aims to represent ship dynamics by constructing its parametric representations and determining the parameters. Over the years, various physics-based models have been proposed for various application [10], such as one degree of freedom (DoF) Nomoto model [11], three DoF Abkowitz maneuvering model [12], the dynamic-positioning model, the vectorial representation [13], and the mathematical modeling group (MMG) model [14].

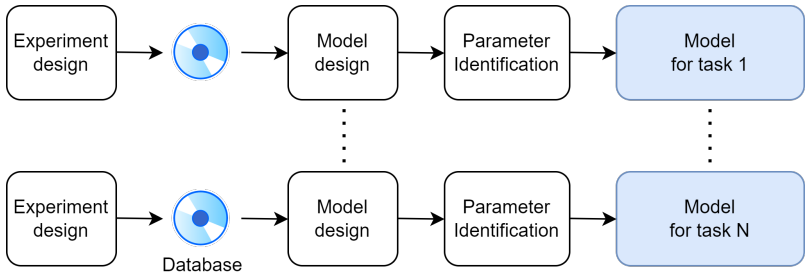


Figure 1.2: A schematic overview of the traditional framework for identifying ship dynamic models for different tasks.

The term "physics-based" is used to describe these models as they place a great emphasis on a mechanical understanding of the phenomenon. Fig. 1.2 illustrates the procedure of the development of a physics-based model for ship dynamics. Although the physics-based model prioritizes physics laws, it does not directly solve the governing Navier-Stokes equations but rather be a statistical surrogate model. Therefore, it is not fully data-free and requires numerical and model-scale experiments to be calibrated. The international towing tank conference (ITTC) recommends that hydrodynamic coefficients of a physics-based model should be validated by benchmark data from model-scale experiments [15]. Free-sailing data from full-scale tests is usually not regarded as a reliable source for model identification due to the lack of force measurements. It is well-known that the design of experiments and models greatly influences performance and robustness of the model. In the 1970s, a model development project for the VLCC "Esso Osaka" was launched [16]. They conducted a blind test with numerous research institutions for letting them build their own maneuvering model for Esso Osaka. The results showed that the developed models varied depending on the development procedure and research institution. Thus, the uncertainties in physics-based models cannot be avoided even by experienced professionals. In addition, physics-based models are surrogate models which are not globally but locally available. They are derived through many simplifications and assumptions. Thereby, the applicability domain of the developed model is determined by operational, environmental, loading, and geographical conditions. This is the "task" that the model targets. As shown in Fig. 1.2, different models need to be built for different tasks.

In this way, physics-based models play a guiding role in showing us the path to understanding phenomena. They serve as a knowledge foundation where we place our understanding and reliability, however, their performance suffers from model uncertainty. In this dissertation, a physics-based model is referred to as **the foundation model** when we focus on its role as a knowledge foundation. The former term focuses on the form of the model while the latter term focuses more on the characteristic of the model.

1.1.2 Data-driven models

A counterpart to the physics-based model is a data-driven model, which models hidden patterns inductively by analyzing and learning from data. It is enabled by data-driven

models being able to learn nonlinear and complex input-output relationships without requiring rigorous understanding of the underlying phenomena. They readily capture nonlinear and complex phenomena of ship dynamics where physics-based models require great effort to do so. In addition, in recent years, onboard data collection has become more practical due to advancements in sensing and communication technologies. For instance, Kongsberg Maritime offers an easy-access platform for sharing onboard data[1]. ShipDC offers an open platform where stakeholders collect and distribute their data[2]. Such data collection and utilization are a crucial focus area in the maritime industry, and this trend makes data-driven models more popular in this field [4]. For supervised learning, popular data-driven models include neural network (NN) [17], decision tree (DT) [18], Gaussian process regressions [19], Autoregressive models [20], support vector machine (SVM) [21], etc.

The benefits of using data-driven models come with drawbacks. Data-driven models prioritize achieving high performance in the given dataset, thereby, its internal mechanism is usually not clear [22]. They may behave in a way that we do not accept from a domain-knowledge perspective. It is highly demanding for us to make any decisions based on such models by placing our understanding and reliability. Moreover, it does not strike us as a smart idea to learn the fundamental part of the phenomena, where physics-based models may prevail, in an inductive manner. Therefore, in practical applications, data-driven models without any benefits from the knowledge foundation would not be sufficient to fulfill the role of ship dynamic models although they may achieve excellent performances in academic papers.

1.1.3 Data-driven enhancement to foundation models

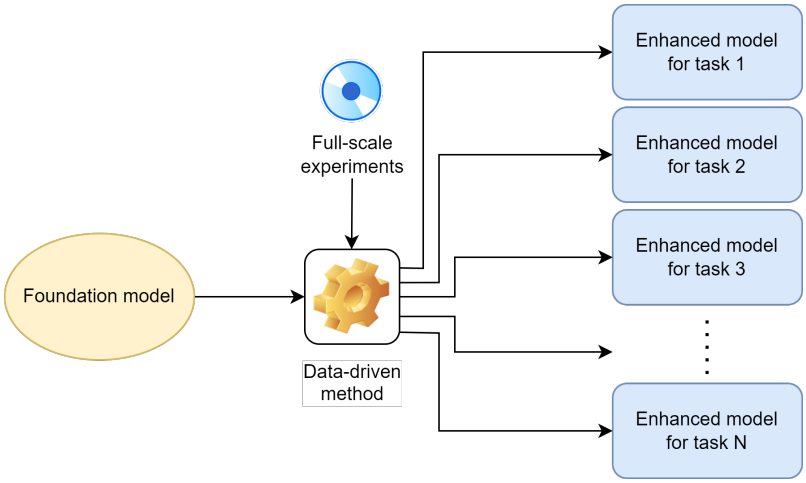


Figure 1.3: A schematic overview of the proposed framework.

[1]Kongsberg digital "Vessel Insight", https://kongsbergdigital.com/products/vessel-insight/, Data accessed 30-March-2023

[2]ShipDC, https://www.shipdatacenter.com/, Data accessed 30-March-2023

Practically, data-driven models may be not considered to be eligible for serving as ship dynamic models on their own, as they do not aim to respect our domain knowledge. However, given that the maritime industry is entering "the era of digitalization" [23], it still seems increasingly important to quickly benefit from full-scale operational data obtained from state-of-the-art sensing and monitoring technologies. To offer dynamic models for ship motion prediction, it is key for data-driven models to support the foundation model to achieve better performance, while still forwarding knowledge from the foundation model. This approach expects a supportive role from data-driven models to maximize the value of the available foundation model as a knowledge foundation.

When building a ship dynamic model in a traditional way as shown in Fig. 1.2, weak knowledge connections between tasks are seen. In fact, fundamental knowledge may be able to be shared between various models for various tasks. Not doing so would require rebuilding that part from scratch, leading to additional costs and risks. Instead, humans usually emphasize knowledge connection. We enhance our understanding by concentrating our efforts into learning essential differences between tasks while making use of existing foundations as they are. In this dissertation, for delivering models in a more agile manner towards various tasks, we emphasize such a process of humans adapting to a new task.

To this end, this dissertation presents a new framework: the data-driven enhancement to the foundation model, which is referred to as **the DE framework** throughout this dissertation. Fig. 1.3 shows a schematic overview of the DE framework. The DE framework assumes that we have a foundation model and a newly-obtained dataset for a targeting task. Given that maritime stakeholders embody their experience and knowledge in the form of validated physics-based models, this assumption is highly pragmatic. For example, taking a physics-based model for ship A as the foundation model, we might want to enhance its performance:

- towards operations which have not been well-trained for.
- with a full-scale dataset.
- towards a similar-but-different ship B.

Instead of newly building a model from scratch, the DE framework aims at modifying, converting, or expanding the foundation model by applying data-driven models with the newly-obtained dataset. This framework enables models to benefit from newly-obtained data, while being anchored to the foundation model. The DE framework is expected to bring the following benefits, but not limited to these, and promote more agile development of ship dynamic models than the traditional framework:

Knowledge transfer; In the DE framework, the foundation model offers a knowledge foundation. This part does not need to be re-discussed and re-learned although the foundation model does not present satisfactory performance for the targeting task. Having a knowledge foundation allows us to transfer useful knowledge and concentrate our efforts on the aspects in which we are interested. This is expected to contribute to better performance that can be easily and consistently produced.

Knowledge continuity; The development of the foundation model and its enhancement

are a chain of the process. We can keep expertise devoted to building the foundation model accessible and useful during the development of the new model. For example, if a strong enhancement seems to be necessary, a team familiar with the foundation model can examine if such enhancement is reasonable by discussing the reason why the foundation model cannot represent such aspects.

Reduce the risk of low performance; When building a new model, there exists no guarantee that it outperforms the existing models. By using such models as a foundation, we might have less risk of deteriorating it as we put much effort into improving it instead of learning it from scratch.

Easy validation & deployment; An enhanced model is built upon a foundation model, which has been validated and widely used. A team engages in a deep discussion of the changes introduced by the data-driven enhancement. Thereby, we can verify the changes and agree on the validity of such changes more easily compared to discussing without any certain baselines.

1.1.4 Research questions

The focus of this dissertation is to shape, explore, and validate the DE framework for building ship dynamic models in a more agile and efficient manner. To this end, the problem awareness needs to be clarified. It prompts the first research question (RQ):

RQ1 **What are challenges when building a ship dynamic model without having a foundation model?**

By reviewing and summarizing the state-of-the-art practices of modeling techniques, this dissertation must answer this research question. The principles of traditional physics-based models must be revisited. This research question leads to the next research question:

RQ2 **How can the DE framework be realized?**

As discussed earlier, the DE framework is expected to bring some benefits, but there would be various ways to achieve it and the expected benefit is achieved to various extents. Principles of data-driven models enabling such enhancement must be discussed. The way we realize the DE framework is likely to be constrained by the project's assumptions and how we value each expected benefit. This dissertation offers their overview. Some previous studies may be related to or even be a part of the DE framework, which must be properly incorporated in this overview. The characteristic of each enabler should also be well-captured. In this dissertation, it should be noted that our focus is not on comprehensively validating such enablers, but rather (1) proposing methods for overcoming high-priority problem settings, and (2) addressing important research questions when implementing the DE framework. It raises the next research question stemming from the high-priority problem settings:

RQ3 **How does the DE framework benefit building a model from a full-scale dataset?**

We are interested not in the model-scale but in the full-scale ship motions. If full-scale data can be effectively reflected in model performance, we can avoid scale effects and significantly reduce the labor-intensive model-scale and numerical experiments, which could be seen as a great benefit. However, in practice, the following challenges are known to do so:

- Full-scale data may be taken under the great influence of external disturbances, which may ruin the identified model.
- Such as docking operations, there exist situations where our domain knowledge does not sufficiently represent the real-world complex ship dynamics in the parametric model.
- Forces and moments cannot be measured directly in the full-scale experiments, thus the quality of measurements is low.
- Limited cases are conducted in full-scale experiments due to time and safety reasons.
- If using existing operational data, the imbalanced distribution of data can have a negative impact on the identified model.

Overcoming these challenges, it is a high-priority task that the DE framework must address to deliver a rigorous model in a timely manner towards full-scale operations where we focus on.

The DE framework may not only bridge the model-scale and full-scale dynamics, but it could also bridge the dynamics of similar ships. It leads to the next research question:

RQ4 **How does the DE framework build a knowledge connection between similar ships? How does it benefit?**

If data cannot be obtained in satisfactory quality and quantity, new ship models may fail or be greatly impacted by perturbations in the dataset, making it difficult to evaluate its validity. In such cases, we might want to introduce a foundation model from similar-but-different ships, thus bringing better performance and stability. By sharing the same foundation model among similar ships, it may be possible to create a "family of ship models" which would bring knowledge connections between tasks.

The next research question naturally arises when considering how the DE framework works in practice:

RQ5 **How do the foundation model and data benefit the performance of the enhanced model? Can we compromise on either of them?**

The DE framework discussed in this dissertation is not simply a fusion of physical knowledge and data-driven models for achieving high performance. Rather, it is for enhancing a particular foundation model using a particular dataset, while still taking advantage of knowledge transfer and continuity from the foundation model. Then, can any model be eligible for being a foundation model for any task? Can any combination

of the foundation model and dataset produce promising results for any task? "Yes" to these questions is highly counterintuitive. This research question needs to be addressed to investigate the basic mechanism of the DE framework.

1.2 Scope of work

1.2.1 Research objectives

To answer the research questions discussed above, this dissertation concentrates on the following Research Objectives (ROs):

RO1 **Investigate how the DE framework, which enhances a foundation model with a data-driven model, can be realized. Clarify how such enablers solve problems that the traditional framework has.**

The goal of this dissertation is to form and validate the DE framework that bridges the foundation model towards a target task by bringing a data-driven enhancement. To achieve this, we need to understand how the DE framework can be realized (corresponding to RQ2) and how they can solve the challenges now we have (corresponding to RQ1). Chapter 2 is devoted to satisfying RO1. Each aspect of the overall picture is discussed in the publications that comprise this dissertation. Associated publications paper i and iii explore the state-of-the-art of existing non-DE framework. This may be relevant to RO1 as it would highlight the challenges that the DE framework can contribute to.

RO2 **Propose techniques that agilely reflect a full-scale dataset into model performance in the DE framework.**

It is the primary interest of the DE framework to readily benefit the model performance from a full-scale dataset, while avoiding challenges in the full-scale dataset as discussed in RQ3. In Paper II and III, the DE framework for full-scale maneuvering is presented. In Paper VI, we shift the focus from maneuvering to seakeeping ship motion with an imbalanced full-scale dataset. In Paper I, the author tackles a more pragmatic problem setting in full-scale real-life maneuvers. In the full-scale dataset, thruster commands could be complexly changing. Paper I presents a new DE architecture that effectively handles such a change both during the training and inference phases, while highly segregating the contributions of the foundation model and data. This is a key solution that accommodates both effectively learning from real-life data and such segregation in the DE framework.

RO3 **Propose a technique that bridges the dynamics of two ships. Explore how one model benefits model identification of the other.**

As pointed out in RQ4, the available foundation models may be often for similar-but-different ships. In such cases, we want to quickly, easily, and reliably identify a model for the targeting ship using such foundation models. Alongside RO3, this must be the primary interest of the DE framework. Paper iv proposes the knowledge transfer strategy using NNs bridging two ships' dynamics. Paper V explores their impact on enhanced performance more comprehensively with numerical and full-scale experiments.

RO4 **Examine how the foundation model and data benefit enhanced performance of the enhanced model.**

The twin pillars of the DE framework are the foundation model and data. When considering the DE framework, we usually aim to compromise on both. If the foundation model must present rigorous performance to the target task, the DE framework is not feasible as such a model is rarely found. Otherwise, the modeling effort can be significantly reduced by using a compromised model fulfilling a role as the foundation model. When it comes to data, it is extremely difficult to obtain a large amount of onboard experimental data. Thereby, it is key to be well aware of how the foundation model and data impact enhanced performance of the enhanced model. Paper ii shows an initial case study that examines the contribution of the foundation model and data to performance. Subsequently, this issue is profoundly discussed in Paper IV with large-scale experiments.

1.2.2 Interconnection between the research objectives

The interconnection between the ROs and the published papers are shown in Fig. 1.4. Paper i and iii are associated publications that the author contributed as a co-author. They investigate physics-based and data-driven principles, which highlights the necessity of the DE framework. All publications contribute to achieving RO1 in a way that investigates and realizes the DE framework on different experimental platforms. RO4 addresses a key question about the mechanism of the DE framework. Paper ii presents a preliminary work where physics-based, data-driven, and enhanced models compete and are compared. In Paper IV, employing the enhanced model inspired by the work in paper I, extensive experimental works is presented to investigate how the foundation model and data benefit enhanced performance.

The rest of the published and associated papers is dedicated for proposing new techniques for enabling the DE framework, focusing on pragmatic problem settings. Related to RO2, four case studies, paper I, II, III, and VI, showcase how the DE framework benefits the foundation model from full-scale data, presenting its novel architectures. Related to RO3, paper V presents a Bayesian framework that bridges dynamics of similar ships for obtaining a new model for a new ship in a more efficient and stable manner. In paper iv, neural networks are employed for addressing RO3 for bridging two ships' dynamics.

1.3 Contributions of the dissertation

The major contributions of this dissertation are as follows, which correspond to the ROs above:

- Present a framework to use data-driven models for enhancing the physics-based model serving as a knowledge foundation. Clarify its overview and possible enablers. It is related to RO1.

- Implement techniques of the DE framework for rapidly reflecting the full-scale data to the better performance while utilizing a knowledge foundation from the foundation model. It is related to RO2.

- Propose a novel method for bridging similar ships' dynamics for identifying model parameters more easily and stably while establishing a knowledge connection. It is related to RO3.
- Investigate how the foundation model and data benefit enhanced performance in the DE framework. Extensive experimental study is presented, related to RO4.

1.4 Structure of the dissertation

This introductory chapter described the background and motivation for this dissertation, clarifying its research goals through research questions. The rest of this dissertation unfolds as follows. Chapter 2 introduces the principles of building the foundation model and data-driven models, which are pillars of the DE framework. Related to RO1, an overview of its enablers will also be discussed. In addition, Chapter 2 presents experimental platforms for simulation and full-scale experiments in this dissertation. Chapter 3 presents techniques for realizing the DE framework with reflecting full-scale dataset, including research results from paper I, II, III, and VI. Their applications are ship motion prediction for maneuvering and seakeeping operations. Chapter 4 experimentally investigates how the foundation model and data benefit enhanced performance through extensive simulation and full-scale experiments. Chapter 5 directs a spotlight towards bridging dynamics of similar ships related to RO4, including research results from paper V. Chapter 6 concludes this dissertation, summarizing the contributions and possible future works.

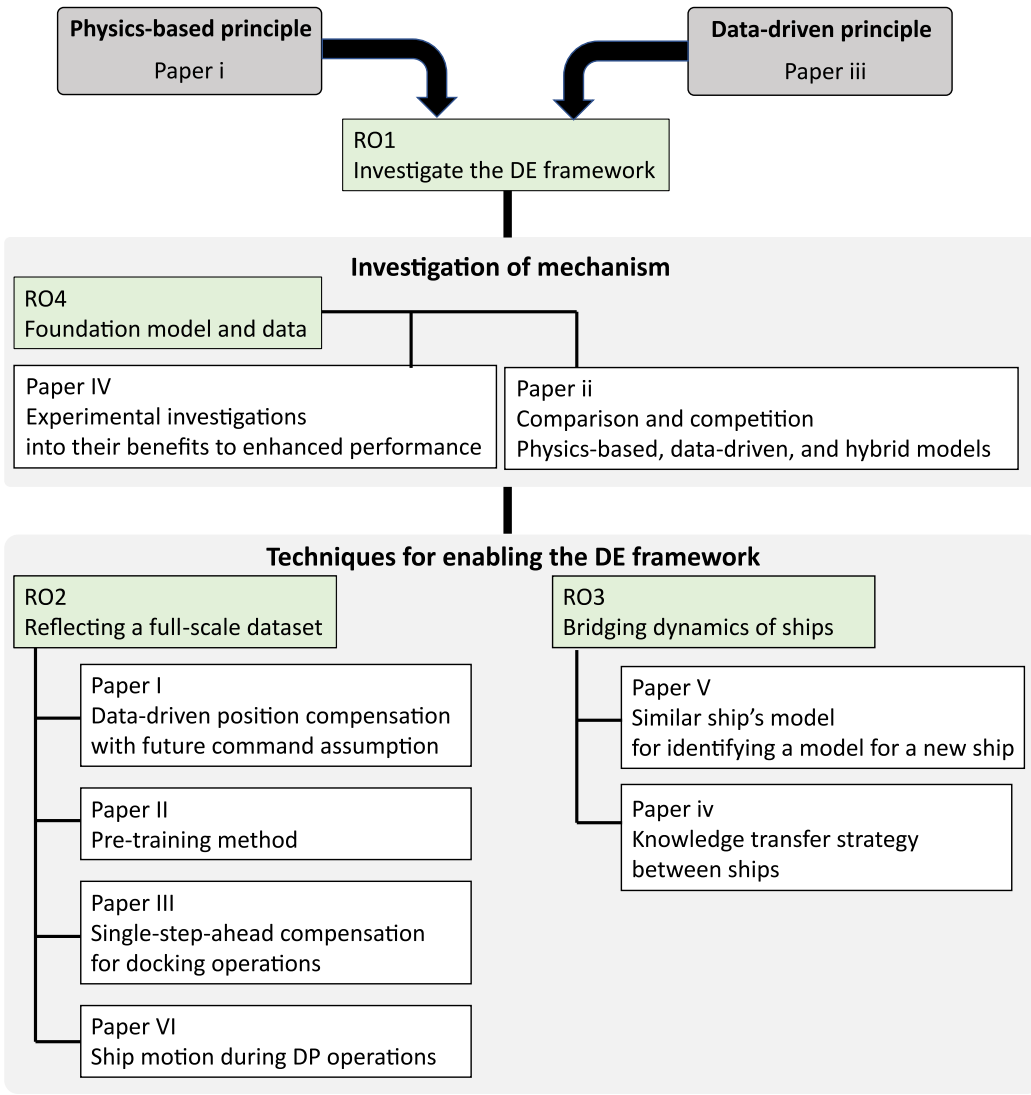


Figure 1.4: Overview of the interconnections between the research objectives and publications. Papers notated with the capital Roman number indicate publications that this dissertation is mainly based on. Those with the lower Roman number indicate associated publications the author contributed as a co-author.

2

Data-driven enhancement to the physics-based model

This chapter explores the overview of enablers of the framework: the data-driven enhancement to the foundation model (the DE framework). They are enabled in various ways, resulting in appreciating the expected benefits of the DE framework to different extents. Fig. 2.1 illustrates how the DE framework readily offers ship dynamic models for ship motion prediction, which is the bedrock for making SA and onboard decision makings.

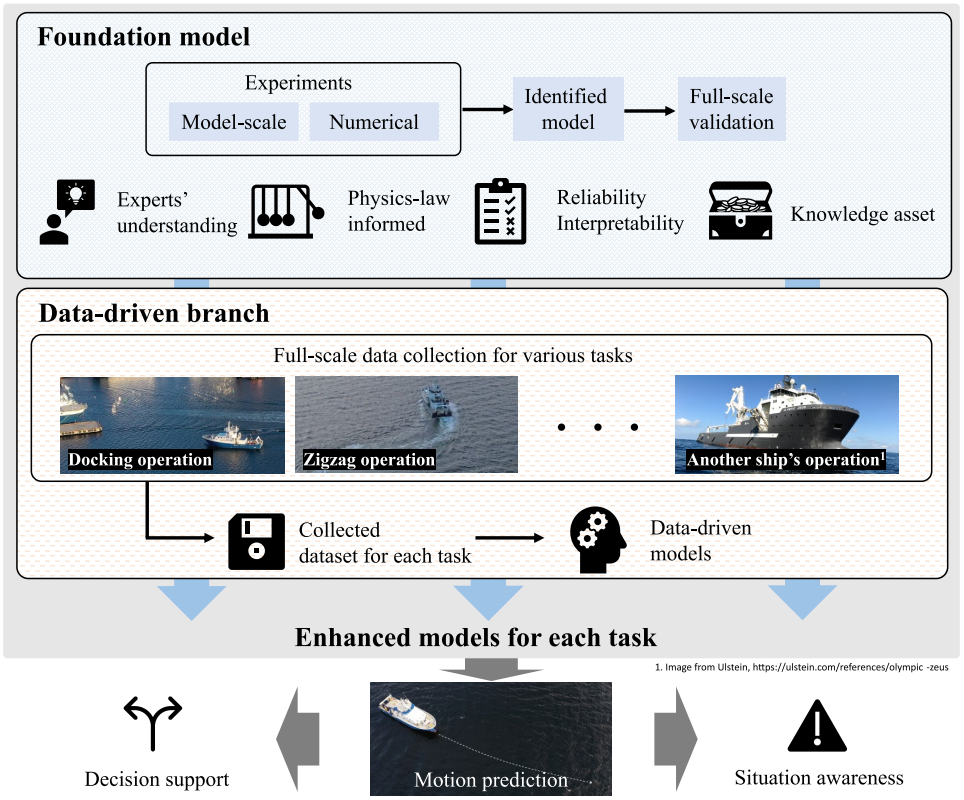


Figure 2.1: Illustration of the data-driven enhancement of the foundation model.

The motivation of the DE framework is to rapidly benefit the existing knowledge

asset from a newly-obtained dataset for better predictive performance. Section 2.1 proposes an overview of enablers of the DE framework. In Section 2.2, the principles of developing the foundation model in a physics-based way is discussed. The author summarises the principles of data-driven models for enhancing the foundation model in Section 2.3. Simulation and real-world experiment platforms are described in Section 2.4.

2.1 Data-driven enhancement to the foundation model

As discussed, a ship is a comprehensive system involving multi-scale and multi-disciplinary sub-systems such as surrounding water, a hull, engines, and thrusters. Moreover, capturing ship dynamics becomes more difficult due to the fact that such a comprehensive system is exposed to complex disturbances, such as waves, ocean current, and wind. In such settings, we need two different principles to be emphasized for rapidly delivering a dynamic model. One is a solid knowledge foundation, shown in the blue panel in Fig. 2.1, where experts have a good understanding and place their reliability. It is represented by the physics-based model developed through performing careful experiments and appreciating domain knowledge, which serves as the foundation model in the present framework. The other is data-driven branches stemming from and based on the foundation model, which enables rapidly adapting to the newly-obtained dataset, shown in the orange panels in Fig. 2.1. This branch concentrates on being rapidly built up by learning from datasets collected by modern ships equipped with diverse sensors. Multiple branches can be developed for various tasks while sharing the same foundation model, which enables knowledge transfer and continuity. The advantages and disadvantages of the two principles are shown in Tab. 2.1. In this section, we focus on capturing the functional differences between possible enablers and over-viewing them through categorization.

2.1.1 Related works in interdisciplinary fields

As a higher-level concept including the DE framework, the synergy of scientific knowledge and data-driven modeling can be considered. Through interdisciplinary fields, such efforts have been widely popular. Rueden et al. [24] called some of such efforts the informed ML that incorporates not only scientific knowledge, but also general knowledge of learning systems. They organized realization methods with three perspectives: knowledge source, knowledge representation, and knowledge integration. Karpatne et al. [25] more focused on the synergy of scientific knowledge and data, which is called theory-guided data science. They believe that model performance should be evaluated in a way:

$$\text{Model performance} \propto \text{Model accuracy} + \text{Model simplicity} + \text{Physical consistency} \quad (2.1)$$

This belief seems reasonable as hypotheses should be taken from a physically reasonable hypothesis space. They proposed five categories for implementation:

- Deciding the model structure based on scientific knowledge.
- Learning the model with the help of scientific knowledge.
- Refining the model outputs based on scientific knowledge.
- Building hybrid models of scientific theory and data science.

Table 2.1: Advantages and disadvantages of physics-based and data-driven modeling principles.

Principle	**Advantages**	**Limitations**
Physics-based	• derived from physics principles. • presents high extrapolation performance, if modeling assumptions are valid. • compromised models may be easily derived. • experts place good understanding and reliability on how real-world phenomena are described in the model.	• requires much expertise. • requires extensive cost & time to build high-performance model. • does not handle highly complex and nonlinear phenomena.
Data-driven	• requires less expertise. • can quickly adapt to a dataset with high performance. • access to state-of-the-art ML models. • robust to noise and disturbances.	• lack of interpretability. • does not necessarily respect physics principles. • requires large datasets.

- Data augmentation based on scientific theory.

[26, 27] extensively reviewed the physics-informed ML, which integrates physics principles into ML. These methods have been particularly successful in chemical engineering [28], biology [29], and fluid dynamics [30]. Transfer learning" [31, 32] is a more general concept of transferring knowledge from a source to a target domain, which may partly cover such a synergy from an implementation perspective.

These concepts explore the synergy of scientific knowledge and data-driven methods. The level of their cooperation and methodology vary. In contrast, the focus of this dissertation is to enhance the performance of the foundation model, that is packaged holistic knowledge, rather than incorporating incomplete knowledge into data-driven models. The DE framework for obtaining a ship dynamic model could be inspired by such concepts from the implementation perspective. However, there is no domain-agnostic method that works best for any application in any field. It is because both scientific knowledge and data are something domain-specific, at the same time, their synergy always needs to take domain-specific constraints into account. This argument motivates us to conduct a domain-specific investigation of enabling the DE framework in the field of ship dynamics. As a first step, in this chapter, we categorize the possible

enabler of the DE framework into four modes for this domain.

2.1.2 Four modes of enabling the DE framework

This subsection defines four modes which determines how the foundation model relates to its data-driven branch.

The update mode

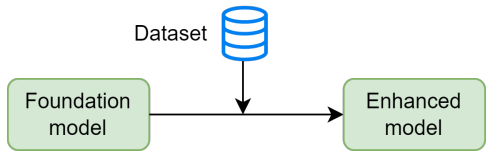


Figure 2.2: The update mode.

The update mode shown in Fig. 2.2 is the most straightforward way to incorporate insights from a newly-obtained data into the foundation model. In this mode, we update the parameters of the foundation model by using the newly-obtained dataset such that it better fits the dataset. The enhanced model is described as:

$$y = f(x|\theta + \Delta\theta) \tag{2.2}$$

where x is the input vector, y is the output vector, and f is the nonlinear function of ship dynamics. θ is a set of parameters of the foundation model and $\Delta\theta$ is its data-driven update. It corresponds to the Bayesian inference, which estimates a posterior distribution by updating a prior distribution with upcoming data. The update mode is clearly distinguished from the other modes since it does not involve any data-driven model in the enhanced model, which can be used in exactly the same way as the foundation model. Such Bayesian frameworks are seen in some previous works for building a ship dynamic model. The Kalman filter (KF) is the Bayesian approach leveraging knowledge represented by a dynamic model and data coming from observation [33–35]. Non-informative or knowledge-driven distribution is taken as an initial estimate of a prior distribution. Alexandersson et al. [36] proposed a novel approach to interweave knowledge and data towards identifying a dynamic model with noisy data. They denoised signals with a preliminary maneuvering model by KF, and then used that denoised signals for identifying a more sophisticated model. Repeating this procedure, they build up a new model by lifting up the preliminary model by using data.

Hence, the update mode would be perfect for finding a robust-and-accurate model by parsimoniously updating the foundation model. Experts can introduce their domain knowledge into decision makings if updates in parameters are justified and acceptable. On the other hand, the update mode is not preferable for adapting to highly complex operations. It is parameter-identification problem of the parametric model, which does not efficiently adapt to complex phenomena. Updating the foundation model must be carried out by experts in a delicate manner, otherwise, the update may ruin the foundation model easily as [37] showcased the failure of the development of physics-based models. In this dissertation, the update mode is explored in paper V for bridging dynamics of two similar ships, transferring knowledge and building mutual connections.

The convert mode

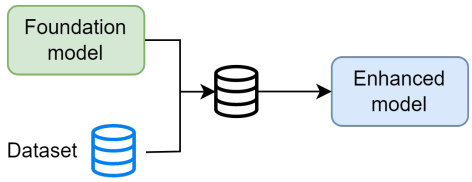


Figure 2.3: The convert mode.

Only the convert mode does not directly embed the foundation model into the enhanced model. There are situations where a designer who develops the enhanced model has no direct access to the foundation model. For instance, a shipping company may need the enhanced model for their SA during their operation, while the foundation model is owned by shipyards or research institutions. Additionally, the foundation model may run only on specific simulation platforms, such as a co-simulation platform or in-house simulator, and may not be easily extracted for embedding purposes. In such cases, simulation results need to be used offline instead of directly embedding the foundation model into the enhanced model. The popular method is to modify the loss function of the learning algorithm such that it regularizes the learning with simulation results or physics constraints. For instance, the loss function can be defined as:

$$\mathcal{L} = \mathcal{L}_t + \lambda \mathcal{L}_s \tag{2.3}$$

$\mathcal{L}$ is the modified loss function. $\mathcal{L}_t$ is the loss function for the discrepancy between the output and target vector. $\mathcal{L}_s$ is the loss function for the discrepancy between the output vectors from the data-driven model and the physics-based model on the simulator. λ is a hyperparameter that balances these two losses. The other way for achieving the convert mode is to augment the dataset with simulators:

$$\mathcal{D}_a = \mathcal{D} \cup \mathcal{D}_s \tag{2.4}$$

where $\mathcal{D}$, $\mathcal{D}_a$, and $\mathcal{D}_s$ are the original dataset taken in the real world, the augmented dataset, and the dataset generated on simulators. Both methods aim to incorporate simulation results into the learning process, rather than solely using real-world data to train the data-driven model. The enhanced model in the convert mode is described as:

$$y = g(x) \tag{2.5}$$

where g is a data-driven function learned from datasets, of which training is facilitated by simulation results. Although the flexibility of the convert mode without requiring embedding the foundation model into the enhanced model is a great advantage, it may become a concern that the foundation model plays an implicit role.

The serial mode

In the serial and parallel modes, the enhanced model has the foundation and data-driven models for data-driven enhancement in its architecture. Such a hybrid architecture

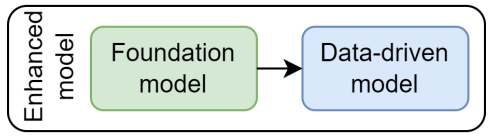


Figure 2.4: The serial mode.

are widely-known as a semi-parametric model [38], a gray-box model [18], or a hybrid model [38]. The serial and parallel modes are structurally different from the update and convert modes, which do not have such two architectures in the enhanced model. In the serial mode, the data-driven enhancement is conducted in a way that the data-driven model is trained such that it maps the output of the foundation model towards the true output:

$$y = g(f(x|\theta), x) \tag{2.6}$$

In [38], it is pointed out that the serial mode has an advantage over the parallel mode when the structure of the foundation model is rather accurate. As the serial mode only maps outputs of the foundation model in a data-driven way, the foundation model plays a more significant role in enhanced performance. In the field of ship dynamics, ensuring the structural fidelity of the foundation model may be difficult, thereby, the advantage of the serial mode over the parallel mode may not be highlighted from the performance perspective. Wang et al. [39] adapted this mode for mapping the dynamics of a ship before the renovation towards that after the renovation. Its interpretability is the biggest advantage since it lifts up the performance of the foundation model only by mapping it. The serial mode can also be defined by reversing the order of the foundation model and the data-driven model:

$$y = f(g(x), x|\theta) \tag{2.7}$$

For instance, Ven et al. [40] used NNs to replace the damping term in the physics-based model of an underwater vehicle. The concept of the reverse serial mode may also include data-driven sensitivity analysis [41] to simplify the foundation model. This reverse mode would fundamentally change the foundation model which is utilized as a packaged knowledge foundation where we place our knowledge and reliability. Thereby, in this dissertation, the reverse mode is not mainly discussed as one of the enablers of the DE framework.

The parallel mode

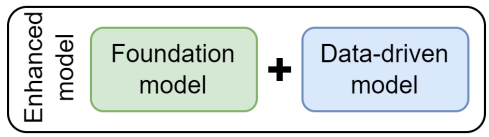


Figure 2.5: The parallel mode.

In the parallel mode, data-driven models are trained such that they compensate for errors made by the foundation model, thus making better prediction in a cooperative way:

$$y = f(x|\theta) + g(x) \tag{2.8}$$

[38] pointed out the parallel mode can outperform the serial mode when the foundation model has a structural mismatch. The parallel mode improves the overall performance of the enhanced model by compromising a part of it as unexplained while keeping the foundation model. It can adapt to the structural mismatch of the foundation model in a more flexible way than the serial mode does. The assumption that the foundation model always has an accurate structure may not always be correct. The structure of the foundation model is determined through simplifications based on our domain knowledge, and there should be some degree of inaccuracy involved. The parallel mode has been popular in plant engineering [42, 43]. In the field of ship dynamics, Mei et al. [18] have made pioneering attempts for the parallel mode. They have used random forest models to correct the acceleration estimation of a physics-based model, and performed a validation study in a simulation environment. Skulstad et al. [44] proposed a method using NN to correct the future trajectory estimated by the physics-based model.

2.1.3 One-shot or iterative

In 2.1.2, the focus is directed on the relationship between the foundation model and data-driven models, and categorized them into four categories. Here, the author emphasizes another key consideration in realizing the DE framework: if the enhanced model is repeatedly used for obtaining the final prediction. This perspective is particularly important when predicting ship motions in the time domain as pointed out in [38, 45]. In the time domain, the final prediction we need is a multiple-step-ahead prediction of ship states in the future.

The iterative approach is more intuitive. The output of the enhanced model is recursively used as the new input to the enhanced model in the next iteration. The enhanced model performs a single-step-ahead prediction repeatedly to obtain the necessary multi-step-ahead prediction over the prediction horizon, in an autoregressive manner. The strength of the iterative approach is that the data-driven branch is learned at a level close to the physical principles, making the learning problem rather simple. It leads to a high learning efficiency for obtaining the data-driven branch, enabling to deal with complex phenomena. Given that the data-driven branch does not work well for input outside the learning range, it becomes a challenge to give sufficient consideration if such conditions are met over the entire prediction horizon in practice. The foundation model and data work in a more interwoven way over the prediction horizon, making them communicate with each other more intensively and efficiently than the one-shot approach does.

On the other hand, in the one-shot approach, the enhanced model runs only once to obtain multiple-step-ahead prediction over the prediction horizon. For example, in the parallel mode, the data-driven model could compensate for multiple-step-ahead errors in trajectories predicted by the foundation model. The advantage of the one-shot approach is that the data-driven branch is strongly separated from the foundation model. It runs only once, so there is less concern for ensuring the data-driven branch runs inside the learning range over the prediction horizon. But, there are also disadvantages. The one-

shot approach requires learning complex problems. Control commands may change over the prediction horizon, thereby, in the one-shot approach, we need to include such high-dimensional command vectors as inputs of data-driven models. These factors result in poor learning efficiency compared to the iterative approach. The two approaches run on different mechanisms and each has its own strength and drawbacks. It is not about one being superior to the other, but rather understanding the drawbacks while focusing on the strengths when using them.

2.2 The principles of physics-based models

A physics-based model can be defined in the time and frequency domains. Although there are various models for various applications, this section presents the time-domain maneuvering model and frequency-domain linear response model employed in this dissertation. The other models can be found in [10]. When a physics-based model serves as a knowledge foundation, this dissertation refers to it as a foundation model. The former is named based on the type of the model, and the latter is named based on the role of the model.

2.2.1 Maneuvering model

The maneuvering model is a ship dynamic model which represents a 3DoF behavior of the ship on the North-East plane. The behavior is described in the surge, sway, and yaw directions. For the rigid-body kinetics, it is formulated as:

$$\begin{aligned} &\dot{\eta} = R(\psi)\nu \\ &M_{RB}\dot{\nu} + C_{RB}(\nu)\nu + M_A\dot{\nu}_r + C_A(\nu_r)\nu_r + D(\nu_r) + D_n(\nu_r)\nu_r = \tau_c + \tau_{wind} + \tau_{wave} \end{aligned} \tag{2.9}$$

where $\eta = [x, y, \psi]^T$ is the ship position vector, which includes north, east positions, and heading in the Earth-tangential North-East-Down (NED) frame. $\nu = [u, v, r]^T$ is the ship velocity vector, which includes the surge, sway, and yaw velocities in the body-fixed frame. $R(\psi)$ is the rotation matrix on the horizontal plane, which relates the NED and body-fixed frame:

$$R(\psi) = \begin{bmatrix} cos(\psi) & -sin(\psi) & 0 \\ sin(\psi) & cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.10}$$

$M_{RB} \in \mathbb{R}^{3\times3}$ and $M_A \in \mathbb{R}^{3\times3}$ denotes the rigid-body and added mass matrices, respectively. $C_{RB}(\nu) \in \mathbb{R}^{3\times3}$ and $C_A(\nu_r) \in \mathbb{R}^{3\times3}$ represent the rigid-body and added Coriolis and centripetal matrices, respectively. $\nu_r = \nu - \nu_c$ is the relative velocity vector through the water with consideration for the generalized ocean current velocity of an irrotational fluid $\nu_c = [u_c, v_c, 0]^T$.

In (2.9), the right-hand side of the second equation for ship kinetics describes external forces and moments acting on the ship. It is composed of the control vector $\tau_c \in \mathbb{R}^3$, which is produced by the actuator systems, the wind vector $\tau_{wind} \in \mathbb{R}^3$, which is produced by wind acting at a certain angle relative to the ship, and $\tau_{wave} \in \mathbb{R}^3$, which is produced by waves. The control vector of forces and moment is calculated by using hydrodynamic surrogate models for actuators. They are usually provided by the manufacturer of actuators or identified through experiments.

During ship maneuvering, the accurate estimation of wave forces and moments acting on the ship is highly challenging due to the lack of sufficient modeling knowledge and measurements. Thereby, throughout this study, it is assumed to be unknown $\tau_{wave} = [0, 0, 0]^T$. Similarly, the accurate onboard measurement of ocean current is mostly not available, resulting in assuming $\nu_r = \nu$ throughout this study. This means that the onboard wind measurement is often the only way to obtain information about external disturbances. Based on the tunnel or numerical experiment results, the wind vector of forces and moment can be expressed as follows:

$$\tau_w = \frac{1}{2}\rho_a V_{rw}^2 \begin{bmatrix} C_X(\gamma_{rw})A_{FW} \\ C_Y(\gamma_{rw})A_{LW} \\ C_N(\gamma_{rw})A_{LW}L_{oa} \end{bmatrix} \tag{2.11}$$

where the relative wind speed is notated as $V_{rw} = \sqrt{u_{rw}^2 + v_{rw}^2}$, its attack angle is $\gamma_{rw} = -atan2(v_{rw}, u_{rw})$. $u_{rw} = u - V_w \cos(\beta_w - \psi)$ and $v_{rw} = v - V_w \sin(\beta_w - \psi)$ are the longitudinal and lateral components of the relative wind speed. V_w and β_w denote the wind speed and its direction, respectively. C_X,C_Y, and C_N are wind coefficients, which are specifically determined for superstructure shape exposed to the wind. A_{FW} and A_{LW} are frontal and lateral projected areas and L_{oa} is the overall length of the ship.

2.2.2 Stochastic model for linear ship response

During DP operations, ships maintain their positions and heading on the two-dimensional plane by using thrusters. Ship's wave-frequency responses in the heave, pitch, and roll during DP operations gain particular interest in ensuring the onboard safety. For such a task, it is common to see a ship as a buoy exposed to waves. Assuming the linear relationship between waves and ship motions, the wave buoy analogy method is popular [46] to relate the energy spectrum of ship motion $S_i(\omega)$ and directional wave spectrum as:

$$S_i(\omega) = \int_{-\pi}^{\pi} |\mathrm{RAO}_i(\omega, \theta_w)|^2 S_w(\omega, \theta_w) d\theta_w \tag{2.12}$$

where ω is the angular frequency, θ_w is the relative direction of waves with respect to the ship's heading, $\mathrm{RAO}_i(\omega, \theta_w)$ is the Response Amplitude Operator (RAO) of the ship for the i-th direction, and $S_w(\omega, \theta_w)$ is the directional wave spectrum. Note that the encounter wave frequency is assumed to be identical to the wave frequency since we focus on DP operations. $S_w(\omega, \theta_w)$ is measured by the onboard wave radar. RAO_i is calculated through hydrodynamic workbenches based on the ship's specifications and geometries, in this study, ShipX[1] is employed. The zeroth-order moment of the ship motion spectrum is:

$$m_i = \int S_i(\omega) d\omega \tag{2.13}$$

Then, $\hat{\sigma}_i^m = \sqrt{m_i}$ is the stochastic estimation of ship motion, which is a standard deviation of ship response.

[1]ShipX, https://www.sintef.no/en/software/shipx/, Data accessed 30-March-2023

2.3 The principles of data-driven models

Nowadays, various data-driven models are being widely used for different purposes, and they are also known as ML models. In this dissertation, the author believes that data-driven models alone do not fulfill the role of a ship dynamic model since it is difficult to introduce a physical interpretation, as shown in Tab. 2.1. Therefore, this dissertation utilizes data-driven models as support to enhance the performance of the foundation model. In this section, the author explains the data-driven models used in this dissertation and the method for optimizing their hyperparameters.

2.3.1 Data-driven models

Linear regression

The linear regression model has been a cornerstone of statistics and ML. It describes a linear relationship between explanatory and response variables. Given a vector of inputs $X^T = (X_1, ..., X_p)$, then the linear regression model predicts the response variable Y via:

$$\hat{Y} = \hat{w}_0 + \sum_{j=1}^{p} X_j \hat{w}_j \tag{2.14}$$

It is convenient to include the constant variable 1 in X such that we include the intercept $\hat{w}_0$ in the coefficient vector $\hat{w}$. Then, the linear regression model is written:

$$\hat{Y} = X^T \hat{w} \tag{2.15}$$

$\hat{w}$ is identified such that it maximizes model performance in the given dataset, which is pairs of an input matrix $\mathbf{X}_{N\times p}$ with each row an input vector and N-vector of the response variable $\boldsymbol{y}$. Here, N is the number of samples in the training dataset. Then, it is known that the optimum $\hat{w}$ is analytically determined such that it minimizes the mean squared error (MSE):

$$\hat{w} = (\boldsymbol{X}^T \boldsymbol{X})^{-1} \boldsymbol{X}^T \boldsymbol{y} \tag{2.16}$$

Ridge regression

When the number of explanatory variables is large and the linear regression model shows multicollinearity, the inverse matrix in (5.11) is hardly calculated. In such cases, the coefficients can be significantly unstable and large due to perturbations in the training dataset, leading to poor accuracy for new samples. This phenomenon is referred to as overfitting the training dataset. One way to avoid overfitting is to use the L2 regularization in ridge regression, where the loss function is to be minimized as:

$$\hat{w} = \arg\min_{w} \sum_{i=1}^{N} (y_i - w_0 - \sum_{j=1}^{p} x_{ij} w_j)^2 + \lambda \sum_{j=1}^{p} w_j^2 \tag{2.17}$$

which can be solved analytically:

$$\hat{w} = (\boldsymbol{X}^T \boldsymbol{X} + \alpha \boldsymbol{I})^{-1} \boldsymbol{X}^T \boldsymbol{y} \tag{2.18}$$

where λ is a hyperparameter determined by the analyst and $\boldsymbol{I}$ is an identity matrix.

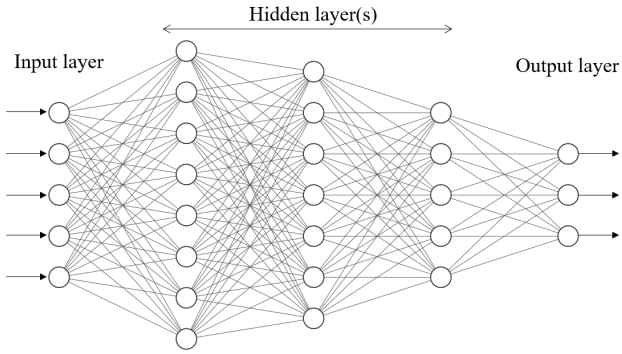


Figure 2.6: An example structure of MLPs.

Multi-Layer Perceptron

Multi-Layer Perceptron (MLP) is the fully-connected feed-forward artificial NN, which is the most classic architecture of NNs. Fig. 2.6 shows an example architecture of MLP. It is composed of a minimum of three layers with nodes in each layer, namely, an input layer, hidden layer(s), and an output layer. The number of nodes in the input layer corresponds to the number of input features. The input layer receives input features and propagates them towards deeper layers. In the hidden and output layers, nodes represent neurons that forward signals from the previous to the next layer:

$$y = a(\sum_{i=0}^{n-1} w_i x_i + b) \tag{2.19}$$

where y is the output signal from the neuron, a is the activation function, x_i is the input signal from the i-th neuron, w_i is the weight of the i-th neuron, and b is the bias. In the hidden layers, non-linear functions (such as sigmoid, tanh, ReLu, and softmax functions) usually used, thus enabling a non-linear mapping from inputs to outputs. The architecture of MLPs is determined by some hyperparameters including the number of layers, the number of neurons in each layer, and the activation function. They must be properly chosen by designers or optimized by the hyperparameter optimization framework.

During training MLPs, we monitor the loss function that indicates the discrepancy between the output and target vectors. The weights and biases in layers are updated by using the backpropagation such that updated MLPs produce a smaller loss.

Long short-term memory (LSTM)

A recurrent neural network (RNN) is a family of NNs, of which output signals from their neurons flow into themselves in a recurrent manner. RNNs focus on better performance for time-series tasks where memory from the past run must be inherited and used in the subsequent run. In particular, Long short-term memory (LSTM) is a type of RNNs [47], which has achieved great success and become the gold standard of RNNs. The highlight of LSTM is that it features in remembering long-term information in time series, while

effectively avoiding the vanishing gradient problem during training. Fig. 2.7 shows an illustration of an LSTM cell.

$$
\begin{aligned}
i_t &= \sigma(W_{ii}x_t + b_{ii} + W_{hi}h_{t-1} + b_{hi}) \\
f_t &= \sigma(W_{if}x_i + b_{if} + W_{hf}h_{t-1} + b_{hf}) \\
g_t &= \tanh(W_{ig}x_i + b_{ig} + W_{hg}h_{t-1} + b_{hg}) \\
o_t &= \sigma(W_{io}x_t + b_{io} + W_{ho}h_{t-1} + b_{ho}) \\
c_t &= f_t \odot c_{t-1} + i_t \odot g_t \\
h_t &= o_t \odot \tanh(c_t)
\end{aligned}
\tag{2.20}
$$

where i_t is the input gate, f_t is the forget gate, g_t is the cell input function, o_t is the output gate, b terms are bias vectors, W terms are weight matrices, c_t is the cell state at time t, h_{t-1} is the hidden state at time $t-1$, x_t is the input at time t, tanh is the hyperbolic tangent function, σ is the sigmoid function $\sigma(x) = 1/(1+e^{-x})$, and $\odot$ is the element-wise product of vectors.

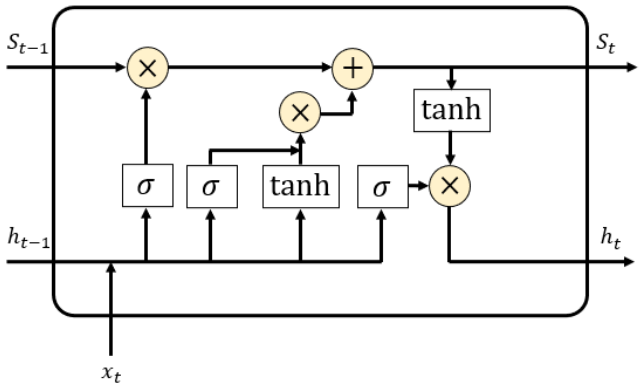


Figure 2.7: Illustration of an LSTM cell.

2.3.2 Hyperparameter optimization

In data-driven models, hyperparameters play an integral role in determining the model architecture and training. They need to be set by the designer instead of being automatically learned from data. For example, neural networks have hyperparameters such as the number of layers, the number of neurons in each layer, and so on. Hyperparameters have a great influence on how data-driven models learn, thus they must be always determined with due consideration.

It is not a good idea to choose hyperparameters such that the data-driven model performs the best in the training dataset. The more complex model can achieve higher performance in the training dataset since it can almost memorize the training dataset, rather than generalizing the underlying phenomena. Such a model may lack generalization performance, and their predictive performance on new samples is low. This pitfall is known as overfitting the training dataset. Thereby, for determining hyperparameters,

we usually keep a part of the training dataset untouched during training. We call this untouched dataset the validation dataset, in which model performance is monitored for choosing hyperparameters. The drawback is the fact that the size of the training dataset is reduced due to keeping the validation dataset from it. To remedy such a drawback, cross-validation is a popular technique. In cross-validation, we split the training dataset into several subsets, where one of them is used for the validation and the others are used for the training. This is repeated by switching the subset for the validation, thus making it possible to make use of the whole training dataset while effectively avoiding overfitting it.

Optuna framework

Optuna [48][2] is an open-source hyperparameter optimization framework, which enables us to efficiently search a set of hyperparameters that maximizes model performance. Optuna is a library-agnostic framework, so it can run with any ML libraries such as scikit-learn[3], tensorflow[4], keras[5] , and PyTorch[6]. In some publications in this dissertation, an optimum set of hyperparameters is searched by using the Tree-structured Parzen Estimator (TPE) sampler [49, 50] in optuna. The TPE sampler is one of the Bayesian optimization methods, which has been widely used for the hyperparameter tuning thanks to its great performance and small computational time.

2.4 Experimental platforms

The focus of this dissertation is building a ship dynamic model for motion prediction. Data on ship motion needs to be taken from experimental platforms for conducting case studies. In this dissertation, the author employs three experimental platforms. The R/V Gunnerus is a real-world ship, which provides a full-scale dataset with diverse onboard sensors. Co-simulation platform and MSS toolbox are well-validated simulation platforms, which enable to conduct high-quality virtual experiments.

2.4.1 The R/V Gunnerus

Figure 2.8: A snapshot of the starboard view of the R/V Gunnerus.

The Research Vessel (R/V) Gunnerus is a multipurpose research vessel operated by the NTNU. She has been mainly used for research and educational purposes. Her

[2]Optuna, https://github.com/optuna/optuna, Data accessed 30-March-2023
[3]Scikit-learn, https://scikit-learn.org/stable/, Data accessed 30-March-2023
[4]Tensorflow, https://www.tensorflow.org/, Data accessed 30-March-2023
[5]keras, https://keras.io/, Data accessed 30-March-2023
[6]PyTorch, https://pytorch.org/, Data accessed 30-March-2023

Table 2.2: Basic specifications of the R/V Gunnerus for the short and elongated versions.

Specification	Short	Elongated
Length overall (m)	31.25	36.25
Length between perpendiculars (m)	28.9	33.9
Breadth middle (m)	9.6	9.6
Draught (m)	2.7	2.7

Table 2.3: Onboard data channels of the R/V Gunnerus.

Sensor	Channels	Unit
GNSS	Latitude	ddmm.mmmm
	Longitude	ddmm.mmmm
	Surge velocity	knots
	Sway velocity	knots
	Course angle	deg
	Speed over ground	knots
MRU	Heading angle	deg
	Heading rate	deg/s
	Roll angle	deg
	Pitch angle	deg
	Heave displacement	m
	Roll rate	deg/s
	Pitch rate	deg/s
	Heave rate	m/s
Thruster	Port thruster rotational speed	%
	Port thruster angle	deg
	Starboard thruster rotational speed	%
	Starboard thruster angle	deg
	Tunnel thruster rotational speed	%
Anemometer	Wind direction	deg
	Wind speed	knots

starboard snapshot is shown in Fig. 2.8[7]. She was put into operation in 2009 with 31.25m length overall. In 2019, she was elongated from the original length to 36.25m length overall. Her basic specifications are shown in Tab. 2.2 [8]. It should be noted that there are short and elongated versions of the R/V Gunnerus. Both of them are used as experimental platforms in this dissertation.

The diesel-electric propulsion system powers the R/V Gunnerus. She is equipped with two Rolls-Royce 500kW Azimuth thrusters and one Brunvoll 200kW tunnel thruster. She presents an excellent DP performance with Kongsberg SDP-11 system enabled by rotating her Azimuth thrusters 360°.

There are diverse onboard sensor channels on the R/V Gunnerus as shown in Tab. 2.3, which are data sources of real-world data in this dissertation. They provide real-time access to her current status. Moreover, sensor measurements are always logged in 1Hz, which enables us to easily replay her history operations. For building a ship dynamic model, measurements coming from a GNSS reviewer, a motion reference unit (MRU), wind sensors, gyros, and sensors for thruster revolution are used. It is

[7]The R/V Gunnerus, https://www.ntnu.edu/oceans/gunnerus, Data accessed 30-March-2023
[8]For further details: The R/V Gunnerus, https://www.ntnu.edu/gunnerus, Data accessed 30-March-2023

interesting to find that the Miros Wavex radar[9] is installed on her X-band navigational radar. It is a virtual wave radar that can be installed on the existing navigational radar without physically installing a new radar. It provides information about surrounding waves in real-time, such as the wave directional spectrum.

2.4.2 Co-simulation

As mentioned, ships are comprehensive systems composed of multi-scale and multi-disciplinary sub-systems. Co-simulation is an enabler for jointly building up a global simulator only by connecting sub-simulators [51]. Sub-simulators are expected to be black-box and independent of each other between communications, thus making it easier for stakeholders to share their sub-simulators without disclosing how it works. Sub-simulators are to be provided by project partners in a format of a functional mock-up unit (FMU), which is based on a commonly-used standard the functional mock-up interface (FMI), for co-simulation. In the maritime industry, The open simulation platform (OSP)[10] provides an initiative for sharing sub-systems FMUs. The entire ship model can be built after easily assembling them, which runs on an entity-component-system (ECS) based co-simulation framework Vico [53][11].

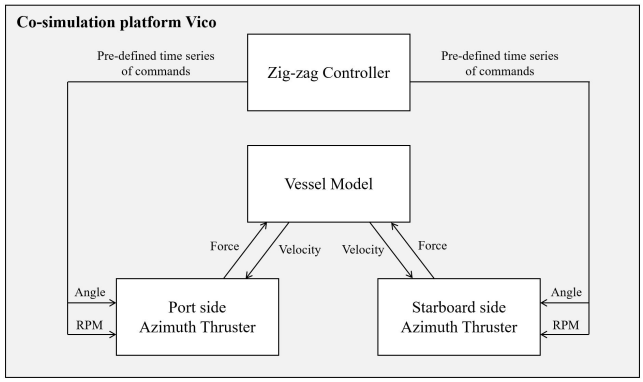


Figure 2.9: A diagram of FMUs and their connections for the entire R/V Gunnerus on Vico.

In this dissertation, we employ a virtual ship of the R/V Gunnerus on co-simulation platform Vico for generating realistic simulation experiments for case studies. In Fig. 2.9, an example diagram of the virtual R/V Gunnerus is shown. Each block represents an FMU for making up the entire ship. The Zig-zag controller is a user-defined controller, which is developed by the author. The controller can be developed in a flexible way by using PythonFMU [54][12] in Python. The Vessel Model is an FMU of a 6DoF ship dynamic model developed by SINTEF Ocean through the SimVal project [55], calculating ship motion under the specified environmental conditions. The Vessel Model is connected

[9]Miros Wavex radar, https://miros-group.com/products/wavex-virtual-sensor/, Data accessed 30-March-2023

[10]The Open Simulation Platform, https://opensimulationplatform.com/ [52], Data accessed 30-March-2023

[11]Vico, https://github.com/NTNU-IHB/Vico, Data accessed 30-March-2023

[12]PythonFMU, https://github.com/NTNU-IHB/PythonFMU, Data accessed 30-March-2023

to two FMUs for Azimuth Thrusters, which are provided by the thruster manufacturer Kongsbergs Maritime. The simulation performance of this virtual R/V Gunnerus has been validated [53].

2.4.3 MSS toolbox

The marine systems simulator (MSS)[13] toolbox is a Matlab-based library for marine systems. Ships, underwater vehicles, and floating structures are available on the MSS toolbox. The MSS toolbox also provides guidance, navigation, and control (GNC) tools. Thereby, users can easily run simulations for different ships on the reliable virtual platform. In this dissertation, two models for two cargo ships (Mariner and SR108) are used for examining knowledge transfer between similar ships in Paper V. Their basic specifications are shown in Tab. 5.1.

2.5 Chapter summary

In this chapter, the author provided an overview of the implementation of the DE framework for developing ship dynamic models. Focusing on the relationship between the foundation model and the data-driven model, we presented four modes enabling the DE framework: namely, the update, convert, serial, and parallel modes. Furthermore, in the time domain, a key perspective to be emphasized is whether to repeatedly use the enhanced model in an auto-regressive manner. The principles of the physics-based and data-driven models was revisited. Finally, we described the three experimental platforms employed in this study.

Tab. 2.4 shows how each paper comprising this dissertation corresponds to these methods and experimental platforms. Through these papers, all four modes are covered. The author prioritizes the one-shot approach over the iterative approach in this dissertation. The former strongly separates the foundation model and data-driven branches, fully respecting the stability of the foundation model. This feature should be widely attractive. In paper III, the iterative approach is taken as it focuses on highly complex docking operations. The update mode runs only in the iterative approach in paper V. Except for paper VI, maneuvering operations are discussed. The R/V Gunnerus in the real world is used as the primary experiment platform.

Table 2.4: Methodologies and experimental platforms in publications in this dissertation.

Paper	**Mode**	**Iterative or One-shot**	**Foundation model**	**Data-driven method**	**Experiment platform**
I	Parallel	One-shot	Maneuvering model	LSTM	Co-simulation
II	Convert	One-shot		MLP	Co-simulation and the R/V Gunnerus
III	Parallel	Iterative		MLP	The R/V Gunnerus
IV	Parallel	One-shot		MLP	Co-simulation and the R/V Gunnerus
V	Update	Iterative		Ridge regression	MSS
VI	Serial	One-shot	Linear response model	Linear regression	The R/V Gunnerus

[13]The Marine Systems Simulator, https://github.com/cybergalactic/MSS, Data accessed 30-March-2023

3 Case study: Enhancement to full-scale operations

The foundation model is a physics-based model that is created based on deep expertise and careful numerical/model-scale experiments. It provides a knowledge foundation that shows the path to understand the phenomenon profoundly, but often, the performance of the model is not satisfactory in the full-scale dataset of interest. This is thought to be due to poor calibration of the foundation model to the dataset and the limitation of the foundation model to fully accommodate the complexity of the full-scale phenomenon. By using such a newly-available full-scale dataset, this chapter presents new methods of enhancing the foundation model with data-driven methods, which is a smarter approach than creating a completely new model or using the foundation model as it is. In this chapter, four case studies are presented to address different realistic problem settings under various constraints. The first three case studies deal with predicting maneuvering motion in the time domain, while the last case study predicts wave-frequency motions in the frequency domain. It covers the contents of paper I, II, III, and VI. Please refer to Tab. 2.4 for methodologies and experimental platforms covered by each paper. Schematic overviews for four case studies are shown in Fig. 3.1, Fig. 3.6, Fig. 3.12, and Fig. 3.17. The enhanced model in each case study is built on two pillars: the foundation model and the data-driven branch. These generate different synergies for different purposes, depending on the case study. For more details on each case study, the appendix could be referred to.

3.1 One-shot enhancement with consideration for command assumption

This section presents research results from paper I. In the proposed method, the foundation model communicates with the data-driven model on the level of the geometry of predicted trajectories. It first uses the foundation model to draw multiple-step-ahead future trajectories and then uses an LSTM, which is good at handling time-series data, for the data-driven compensation of the trajectories. This is the parallel-mode DE framework in the one-shot approach according to the definition in Chapter 2. The key in the proposed method is how to consider command changes over the multiple-step-ahead prediction horizon. In full-scale operation data, there is no guarantee that control commands remain constant over the prediction horizon. Therefore, for both learning and inference, it is important to efficiently consider it. The proposed method demonstrates an efficient way to handle such high-dimensional vectors of command change over the prediction horizon using the foundation model. The validation study was conducted by a virtual ship of the R/V Gunnerus on the co-simulation platform. The virtual operations were exposed to wind, waves, and ocean current with various intensities coming from various directions. Note that in this section, the author uses terminologies that correspond to paper I. Therefore, different terms may be used from those used in Chapter 2

as:

- The model-based predictor corresponds to the foundation model in Chapter 2.
- LSTMs correspond to the data-driven branches in Chapter 2.
- The hybrid predictor corresponds to the enhanced model in Chapter 2.

3.1.1 Methodology

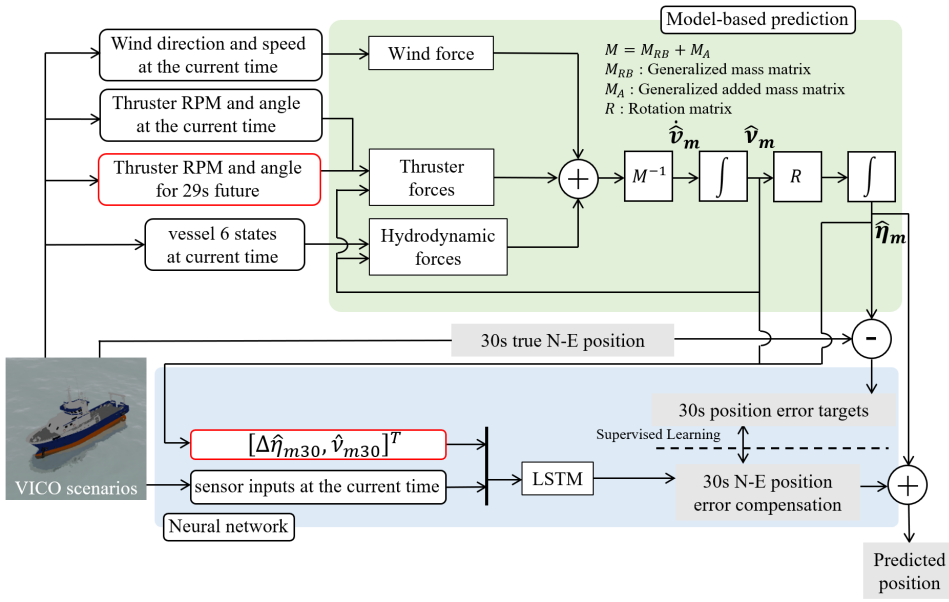


Figure 3.1: A schematic structure of the proposed ship motion predictor with consideration for future command assumption.

Fig. 3.1 shows the proposed structure of the one-shot data-driven enhancement. It is a hybrid architecture of the model-based predictor and the NN which compensates for errors in multiple-step-ahead positions made by the foundation model. This study evaluates 30s prediction performance. The predictor makes 30s North-East position prediction $[\hat{N}_1, ..., \hat{N}_k, ..., \hat{N}_{30}]$ and $[\hat{E}_1, ..., \hat{E}_k, ..., \hat{E}_{30}]$ where $\hat{N}_k$ and $\hat{E}_k$ are predicted North and East positions made by the hybrid predictor at ks future, respectively.

At each time step, the relative wind speed at the current time V_{rw}, the relative wind angle from bow at the current time γ_{rw}, thruster revolution n_0 and angle δ_0 at the current time, a thruster revolution vector $\mathbf{n} = [n_1, n_2, ..., n_{29}]$ and an angle vector $\boldsymbol{\delta} = [\delta_1, \delta_2, ..., \delta_{29}]$ for 29s future, and vessel state $[N_0, E_0, \psi_0, u_0, v_0, r_0]^T$ at the current time are provided to the foundation model where n_k is thruster revolution, δ_k is thruster angle at ks future. In the application of this study, $[\hat{N}_1, ..., \hat{N}_{30}]$ and $[\hat{E}_1, ..., \hat{E}_{30}]$ are calculated to evaluate the decision making $[n_0, ..., n_{29}]$ and $[\delta_0, ..., \delta_{29}]$.

$\hat{\boldsymbol{\eta}}_{mk} = [\hat{N}_{mk}, \hat{E}_{mk}, \hat{\psi}_{mk}]^T$ is a model-based predicted position vector of North, East, and heading at ks future. $\hat{\boldsymbol{\nu}}_{mk} = [\hat{u}_{mk}, \hat{v}_{mk}, \hat{r}_{mk}]^T$ is a model-based predicted velocity vector of surge, sway, and yaw velocities at ks future. $\boldsymbol{\eta_0}$ is a true position vector and $\boldsymbol{\nu_0}$ is a true velocity vector at the current time. The foundation model f_m produces $[\hat{\boldsymbol{\eta}}_{m1}, ..., \hat{\boldsymbol{\eta}}_{m30}, \hat{\boldsymbol{\nu}}_{m1}, ..., \hat{\boldsymbol{\nu}}_{m30}] = f_m(\boldsymbol{\eta_0}, \boldsymbol{\nu_0}, V_{rw}, \gamma_{rw}, n_0, \mathbf{n}, \delta_0, \boldsymbol{\delta})$. A trajectory predicted by the foundation model deviates from a true trajectory due to its low fidelity, the ignorance of the measurement of wave and ocean current, and wave-frequency motion in $[\eta_0, \nu_0]^T$. Therefore, LSTM f_n aims to compensate the error in North and East positions made by the foundation model. The error compensation is described as $[\Delta\hat{N}_{n1}, ..., \Delta\hat{N}_{n30}]$ and $[\Delta\hat{E}_{n1}, ..., \Delta\hat{E}_{n30}]$ in North and East positions, respectively. A function of the error compensation is approximated by the LSTM f_n as $[\Delta\hat{N}_{n1}, ..., \Delta\hat{N}_{n30}, \Delta\hat{E}_{n1}, ..., \Delta\hat{E}_{n30}] = f_n(\sin\psi_0, \cos\psi_0, \boldsymbol{\nu_0}, V_{gw}, \beta_{gw}, n_0, \delta_0, \Delta\hat{\boldsymbol{\eta}}_{m30}, \hat{\boldsymbol{\nu}}_{m30})$ where $\Delta\hat{\boldsymbol{\eta}}_{m30} = [\hat{N}_{m30} - N_0, \hat{E}_{m30} - E_0, \hat{\psi}_{m30} - \psi_0]^T$ with consideration for future command assumption, V_{gw} is the global wind speed at the current time, and β_{gw} is the global wind direction at the current time. ψ_0 is mapped to $\sin\psi_0$ and $\cos\psi_0$ in order to avoid a jumping phenomenon of ψ_0 around 0°. The number of inputs of the NN is 15 and that of outputs is 60. Finally, predicted North position $\hat{N}_k = \hat{N}_{mk} + \Delta\hat{N}_{nk}$ and East position $\hat{E}_k = \hat{E}_{mk} + \Delta\hat{E}_{nk}$ are calculated in the prediction phase. The training dataset is split into five folds for cross validation and hyperparameters are tuned.

The proposed structure makes model-based prediction with consideration for a 29s time series of future command assumption. Subsequently, a model-predicted state only at 30s future $[\Delta\hat{\boldsymbol{\eta}}_{m30}, \hat{\boldsymbol{\nu}}_{m30}]^T$ and sensor inputs at the current time are given to the LSTM. Being different from a straightforward formulation of error compensation by the LSTM $[\Delta\hat{N}_{n1}, ..., \Delta\hat{N}_{n30}, \Delta\hat{E}_{n1}, ..., \Delta\hat{E}_{n30}] = f_n(\sin\psi_0, \cos\psi_0, \boldsymbol{\nu_0}, V_{gw}, \beta_{gw}, n_0, \mathbf{n}, \delta_0, \boldsymbol{\delta})$, an approximation f_n in this study can incorporate the effect of the future command assumption without expanding the number of input variables of the LSTM drastically by making use of model-based predicted future vessel state. By avoiding the curse of dimensionality, it contributes to the high training efficiency.

3.1.2 Validation experiment

Experiment setup

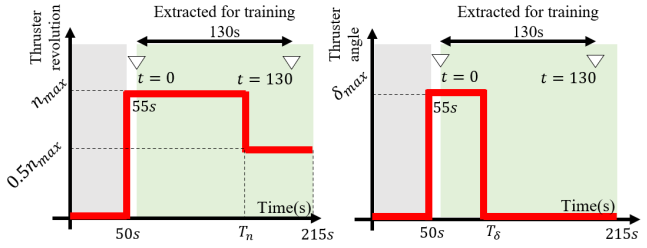


Figure 3.2: Time series of thruster revolution and angle for validation study.

A validation study was conducted by employing a virtual ship of the R/V Gunnerus on co-simulation platform Vico, which enables the generation of virtual full-scale data under environmental disturbances. To reconstruct the deviation of true and model-

based predicted trajectories in the simulation environment, the physics-based model in the model-based predictor was built by introducing uncertainties into the ground-truth model in the simulator. An original scenario is a 215s time history. A 160s time history from 55s to 215s is extracted for an experiment in this study. Hereinafter, the beginning of the extracted scenario is defined as $t = 0$s. As each time step requires a 30s trajectory in the future in the training process, 130s from $t = 0$s to $t = 130$s is defined as one scenario. The virtual R/V Gunnerus is equipped with two azimuth thrusters. The same thruster revolution and angle are given to them in this study. Thruster revolution is set to $n_{\max}$ until $t = T_n$, and then turned back to $0.5n_{\max}$ with the maximum change rate. Thruster angle is set to $\delta_{\max}$ until $t = T_\delta$, and then turned back to zero with the maximum change rate. In the prediction phase, a time series of future command values are given to a predictor. Each scenario is parameterized by a scenario parameter vector $\boldsymbol{P} = [\text{Beaufort wind force scale}, n_{\max}, \delta_{\max}, \theta_d, T_n, T_\delta]$. $T_n \in [20s, 50s]$, $T_\delta \in [80s, 110s]$, $n_{\max} \in [50\text{RPM}, 130\text{RPM}]$, $\delta_{\max} \in [-20°, 20°]$, and $\theta_d \in [0°, 360°)$ are randomly selected for each scenario; then, $n_{\text{all}} = 200$ unique scenarios are generated. The vessel in the scenario is exposed to wind, wave, and ocean current. A calculation of the first-order wave force, the second-order wave force, wind force, and the effect of ocean current are done in the vessel model simulated in Vico by defining the type of wave and wind spectrum, mean wind speed U_w, significant wave height H_s, significant wave period T_s of the spectrum, ocean current speed U_c, and the global direction of environmental disturbances θ_d to the north. JONSWAP spectrum [56] is chosen for the type of wave spectrum. The static wind is applied to the vessel. We generate scenarios under two levels of environmental disturbances based on Beaufort wind force scale[1]. Beaufort wind force scale is level 3: Gentle breeze or level 4: Moderate breeze and defines the combination of (H_s, T_s, U_w, U_c) of the scenario. For the sake of simplicity, θ_d of wind and ocean current is the same as that of the wave. Note that θ_d does not change over time in one scenario. Half of 200 scenarios are under Gentle breeze disturbances and the others are under Moderate breeze disturbances.

Evaluation metrics

We introduce some metrics for plausible evaluations of prediction performance. We have $n_{\text{test}} = 60$ scenarios in the test dataset, which is 30% of n_{all}. We define the prediction error $l_{ijk} = \sqrt{(N_{ijk} - \hat{N}_{ijk})^2 - (E_{ijk} - \hat{E}_{ijk})^2}$ for the scenario i, at a js time step, and at k-step-ahead future. where N_{ijk} is a true North position, $\hat{N}_{ijk}$ is a predicted North position, E_{ijk} is a true East position, and $\hat{E}_{ijk}$ is a predicted East position. We introduce mean prediction error for the prediction horizon $\overline{l_k}$ as $\overline{l_k} = \frac{1}{n_{\text{test}}} \frac{1}{T} \sum_{i=1}^{n_{\text{test}}} \sum_{j=0}^{T-1} l_{ijk}$ where $T = 130$s is the length of a scenario. We look into how a summation of prediction error for 30s changes over time of scenario i as $S_{ij} = \sum_{k=1}^{30} l_{ijk}$.

Results

Fig. 3.3 - Fig. 3.4 show results of a scenario $P = [\text{Beaufort wind force scale} = 4, n_{\max} = 102.2\text{RPM}, \delta_{\max} = -9.0°, T_\delta = 38s, T_n = 96s, \theta_d = 14.4°]$. Time histories of S_{ij} of present hybrid, model-based, and data-driven predictors are shown in Fig. 3.3. In Fig. 3.3, we can see that the prediction performance of the present hybrid predictor outperforms

[1]Met Office, https://www.metoffice.gov.uk/weather/guides/coast-and-sea/beaufort-scale

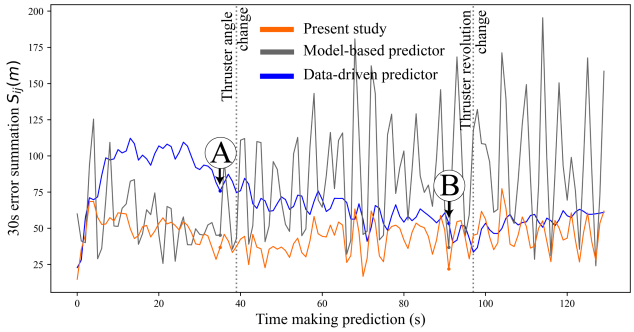


Figure 3.3: Time series of prediction errors S_{ij} made by (orange) the proposed hybrid predictor, (gray) the foundation model, and (blue) the pure data-driven predictor without using the foundation model.

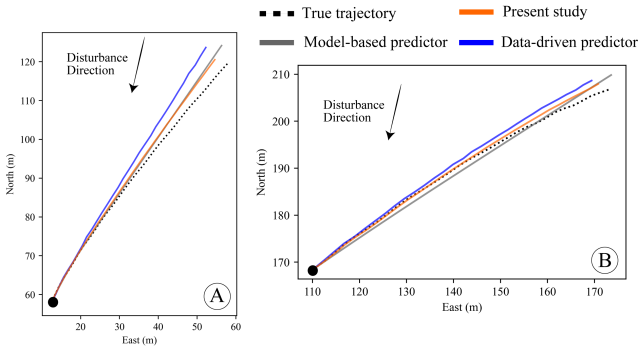


Figure 3.4: Snapshots of the predicted and true trajectories at (A) and (B) shown in the time series in Fig. 3.3.

that of the model-based and data-driven predictors. The model-based prediction stays unstable due to the low fidelity of the model and wave-frequency components in $[\eta_0, \nu_0]$. In order to remove the wave-frequency components from the model-based prediction, one can implement a low-pass filter. However, the tuning of the low-pass filter takes effort and the time delay of the processed signal is inevitable. In the beginning of the scenario, the performances of the hybrid and data-driven predictors are not as good as that of the hybrid predictor after $t = 30$s as they have no sufficient recurrent information. The present hybrid predictor maintains its prediction error at low levels whereas the data-driven predictor produces more error. It should be noted that we do not see the increment of prediction error of the present study around $t = T_\delta$ and $t = T_n$. It indicates that the present hybrid predictor successfully handles a time series of future command assumption while reducing the error made by the low-fidelity model-based predictor.

Snapshots of predictions at (A) $t = 35$s and (B) $t = 91$s are shown in Fig. 3.4. At (A), the vessel turns back the thruster angle from $\delta = -9.0°$ when making prediction to

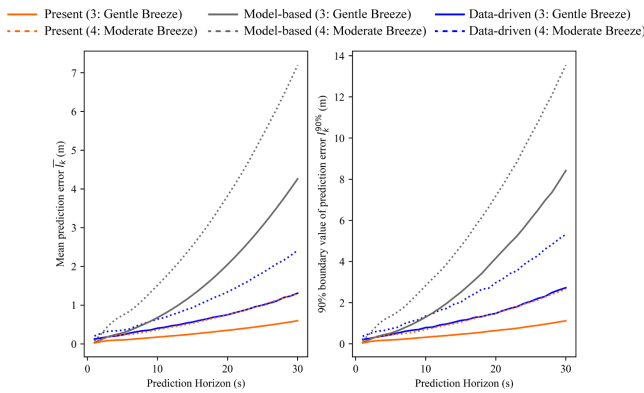


Figure 3.5: Mean and 90% boundary value of prediction error over the 30s prediction horizon in the test dataset.

$\delta = 0°$ at $t = T_\delta$ after making prediction. The dotted line, which is a true trajectory for 30s, shows that the vessel turns clockwise in the beginning of the prediction, however, it gradually goes straight in the last half of the 30s prediction horizon. One can see that the present hybrid predictor captures this behavior and its prediction is more accurate than the other predictors. At (B), the vessel decelerates at $t = T_n$ after making prediction. The model-based prediction ends up with a longer trajectory than the true trajectory, whereas the prediction performance of the present hybrid predictor is very good.

The overall prediction performance in the test dataset is investigated herein. Fig. 3.5 shows the mean prediction error $\overline{l_k}$ in the left panel and the 90% boundary value of prediction error $l_k^{90\%}$ in the right panel over 30s prediction horizon. Solid lines show the prediction performance under lower disturbance (Beaufort wind force scale: 3) and dotted lines show that under higher disturbances (Beaufort wind force scale: 4). In the left figure, LSTM-based predictors (the data-driven and hybrid predictors) outperform the model-based predictor. In particular, the prediction performance of the model-based predictor under the higher disturbance deteriorates significantly whereas its effect on LSTM-based predictors are marginal. In addition, it is discerned that the present hybrid predictor outperforms the data-driven predictor notably especially at the longer prediction horizon. In terms of $l_k^{90\%}$, we see the same trend in the right figure. It indicates that the present hybrid predictor contributes to reducing not only mean prediction error but also the frequency of the large prediction error.

3.2 Enhancement to full-scale docking operations

This section presents the research results from paper III. In this section, we focus on full-scale docking operations where ship dynamics is highly complex and nonlinear transiting from high-speed to low-speed maneuvers. For such operations, the foundation model may not perform satisfactorily as it would not fully capture the complex and nonlinear phenomena in its parametric formulation. In this section, we present the data-driven enhancement to the foundation model by NNs calibrating the single-step-ahead velocity prediction made by the foundation model, thus delivering a ship dynamic model for

docking operations agilely. The proposed method is the parallel mode in the iterative approach. Note that in this section, the author uses terminologies that correspond to paper III. The physics-data cooperative model corresponds to the enhanced model in Chapter 2.

3.2.1 Methodology

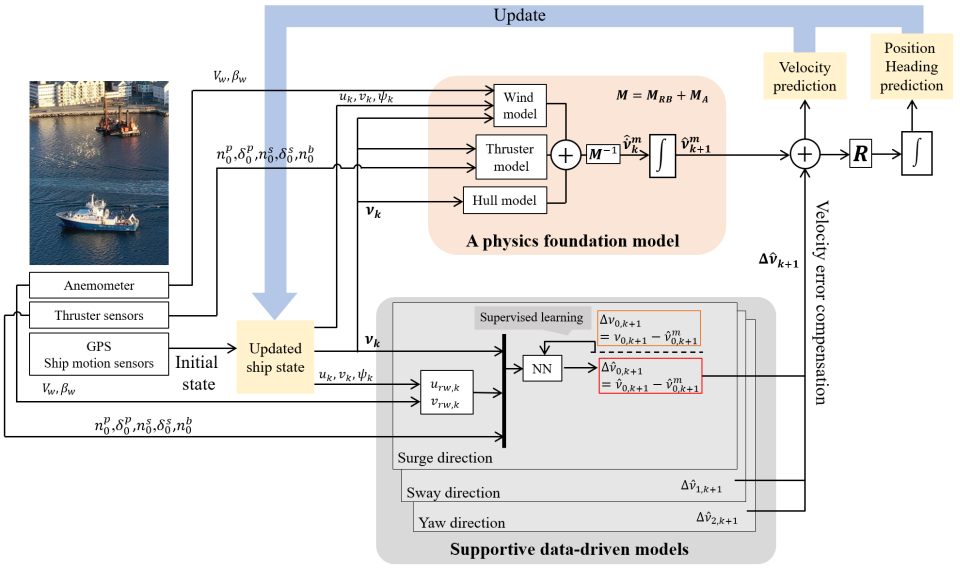


Figure 3.6: A schematic overview of the data-driven enhancement to the foundation model for docking operations.

The schematic overview of the proposed co-operative model is shown in Fig. 3.6. $\boldsymbol{\nu}_0 = [u_0, v_0, r_0]^T$ is three-dimensional velocity vector and $\boldsymbol{\eta}_0 = [N_0, E_0, \psi_0]^T$ is three-dimensional position vector when making prediction. The co-operative model consists of a physics foundation model and supportive data-driven models. Based on onboard sensor measurements from the anemometer, thruster meters, and ship motion sensors, the physics foundation model makes a single-step-ahead prediction of 3DOF velocities. By using onboard sensor data, supportive data-driven models in the surge, sway, and yaw directions serve to compensate prediction errors made by the physics foundation model in each direction. The objective of having supportive data-driven models is to improve the model's poor performance, in the data-driven non-parametric manner, on the kinetic level. To this end, in this study, we employ an MLP, which is one of the representative architecture of ML models, in the supportive data-driven models. Supportive data-driven models in three directions are constructed separately. Hereinafter, suffix d denotes the discrete variables in the surge ($d = 0$), sway ($d = 1$), and yaw ($d = 2$) directions. Suffix k denotes the variables at the time step k ($k = 0$ at the current time). For example, $\nu_{0,k}$ represents the surge velocity at k-step prediction horizon. At k step, the co-operative model makes a single-step-ahead prediction in velocity $\hat{\nu}_{d,k+1} = \hat{\nu}^m_{d,k+1} + \Delta\hat{\nu}_{d,k+1}$

where $\hat{\nu}^m_{d,k+1}$ is a single-step-ahead prediction made by the physics foundation model and $\Delta\hat{\nu}_{d,k+1}$ is an output of the supportive data-driven model. A single-step-ahead prediction is iterated in the autoregressive manner by taking a predicted vessel state at $k+1$ step as a new input of the physics foundation model and supportive data-driven models. Namely, $\boldsymbol{\nu_{k+1}} \leftarrow \boldsymbol{\hat{\nu}_{k+1}}$ and $\boldsymbol{\eta_{k+1}} \leftarrow \boldsymbol{\hat{\eta}_{k+1}}$. The Euler integration method is used for numerical integration.

Input variables of NNs f_{NN} are selected based on the theory of ship dynamics. In the theory of 3DOF ship motion, the velocity of the ship in the next time step is determined by the inertial and hydrodynamic parameters of the ship, the three-dimensional velocities of the ship at the current time, thruster feedback values (revolution and angle) at the current time, and environmental disturbances. In this study, NNs are trained for a specific loading condition of a specific ship, thus the inertial and hydrodynamic parameters of a ship are not included in the input variables. This procedure of feature selection yields:

$$\Delta\hat{\nu}_{d,k+1} = f_{\mathrm{NN}}(\boldsymbol{\nu}_k, n^p_0, \delta^p_0, n^s_0, \delta^s_0, n^b_0, u_{rw,k}, v_{rw,k}) \tag{3.1}$$

$u_{rw,k}$ and $v_{rw,k}$ are discrete values of u_{rw} and v_{rw} at time step k. The number of input variables of supportive data-driven models is 10. Input and target variables are z-normalized by their mean and standard deviation in the training dataset. The same statistic values are applied to the validation and test datasets. Target variables are normalized since normalization contributes to better prediction accuracy in the experiment. In the case study, we further conducted an ablation study of input features. Based on the results of the ablation study, please note that wind information $u_{rw,k}$ and $v_{rw,k}$ are removed from input features in the case study. Hyperparameter tuning is done by hyperparameter tuning framework optuna [48], of which details are found in Appendix.C.

3.2.2 Validation experiment

Experiment setting

The case study employed the R/V Gunnerus's historical data of 88 docking maneuvers acquired from August 2016 to June 2017. 1000s time histories until the truncated time point were saved. 88 docking maneuvers were randomly grouped into the training dataset (56 operations), the validation dataset (14 operations), and the test dataset (18 operations).

Ablation study

For the ablation study, four models with different input features of supportive data-driven models were constructed as follows:

- *(A) No data-driven models*: the physics foundation model makes prediction without the help of supportive data-driven models.
- *(B) Only velocities*: input features of supportive data-driven models include only velocity measurements. Namely, it is formulated as:

$$\Delta\hat{\nu}_{d,k+1} = f_{\mathrm{NN}}(\boldsymbol{\nu}_k) \tag{3.2}$$

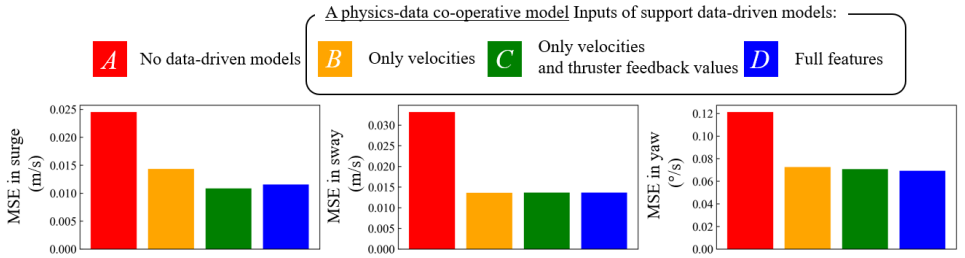


Figure 3.7: Results of ablation studies of input features of supportive data-driven models.

- *(C) Only velocities and thruster feedback values*: input features of supportive data-driven models include only velocity measurements and thruster feedback values. Namely, it is formulated as:

$$\Delta\hat{\nu}_{d,k+1} = f_{\mathrm{NN}}(\boldsymbol{\nu}_k, n_0^p, \delta_0^p, n_0^s, \delta_0^s, n_0^b) \tag{3.3}$$

- *(D) Full features*: supportive data-driven models have a full-form of input features presented in (3.1).

Fig. 3.7 show MSEs of a single-step-ahead velocity prediction in the surge, sway, and yaw directions in the validation dataset. It is clearly seen that, only by having the ship velocities in input features *(B)*, prediction errors made by the physics foundation model *(A)* were significantly reduced in three directions. It shows the effectiveness of supportive data-driven models for compensating prediction error made by nonlinear and complex hydrodynamic forces that were poorly modeled in the physics foundation model *(A)*. In addition, by having thruster feedback values in input features *(C)*, a moderate improvement of prediction performance in the surge direction was seen. Thereby, in the surge direction, to some extent, supportive data-driven models might contribute to compensate prediction error made by a poor fidelity of thruster deterministic models in this dataset. On the other hand, the improvement of prediction performance was not seen by having wind information *(D)* as shown in the blue bars in Fig. 3.7. Thereby, in the case study, we removed wind information from input features of supportive data-driven models. Hereinafter, the formulation of supportive data-driven models follows (3.3). Mostly, we maneuver the R/V Gunnerus under the moderate environmental disturbances due to safety reasons. Thereby, it is reasonable that wind measurement is not informative to improve prediction performance in this dataset.

Prediction for one example operation

Fig. 3.9 and Fig. 3.10 show results of one operation in the test dataset. Time histories of the surge, sway, and yaw velocities of this operation are shown in Fig. 3.8. At the beginning of this operation, the R/V Gunnerus maneuvered straight at high surge velocity. At 400s, she started to turn clockwise and decelerate smoothly. We define the error reduction rate r_e by having supportive data-driven models in the co-operative model as:

$$e_r = (l_p - l_c)/l_p \tag{3.4}$$

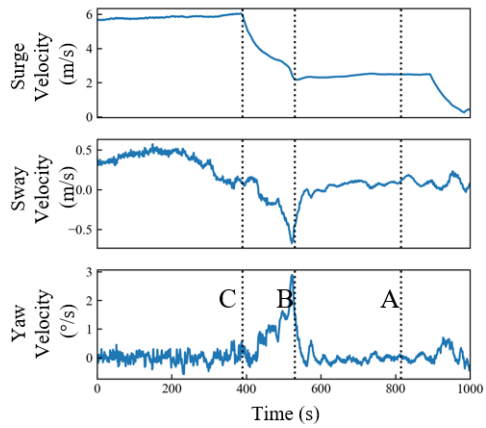


Figure 3.8: Time histories of the surge, sway, and yaw velocities of the example operation in the test dataset.

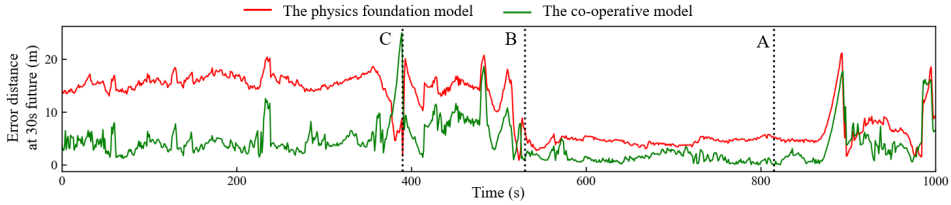


Figure 3.9: Time histories of prediction errors of the example operation in the test dataset at 30s future made by the foundation model and the enhanced model.

where l_p and l_c is a distance error made by the physics foundation model and the co-operative model at 30s future, respectively. In this operation, at time step A ($t = 815$s), the error reduction rate e_r was the highest. Time step B ($t = 530$s), is right after when the yaw velocity was the highest in this operation. At time step C ($t = 390$s), the error reduction rate e_r was the lowest in this operation. In Fig. 3.10, 30s true and predicted trajectories are shown for the time step A, B, and C. At time step A, the R/V Gunnerus maneuvered almost straight for 30s. Whereas the trajectory predicted by the co-operative model traced the true trajectory, that predicted by the physics foundation model was longer and diverged from the true trajectory. It indicates that, at time step A, supportive data-driven models contributed to improving prediction performance, in particular, in the surge and yaw directions. At time step B, the contribution of the yaw-direction supportive data-driven model was clearly seen. The physics foundation model failed at accurately capturing the ship dynamics in the yaw direction, thus ending up with a large deviation from the true trajectory at 30s future. On the other hand, the co-operative model succeeded in tracing the true trajectory precisely and ended up with the almost same location as the true trajectory at 30s future. The intrinsic limitation of data-driven models is that it is inevitable from the aleatoric and epistemic uncertainties.

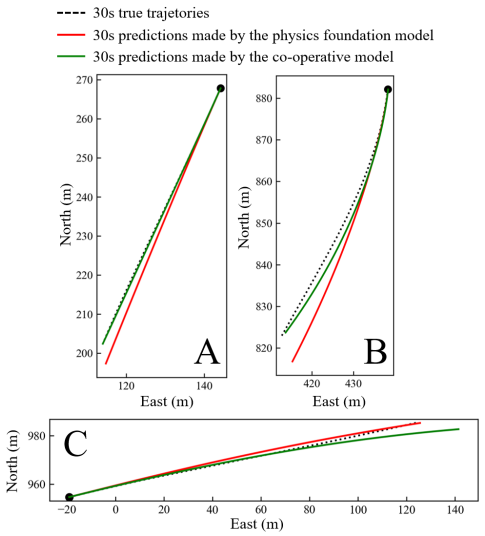


Figure 3.10: 30s true and predicted trajectories at time step (A), (B), and (C) of the example operation in the test dataset shown in Fig. 3.9.

Therefore, supportive data-driven models do not ensure that it always improves the prediction performance of the physics foundation model. At time step C, the predicted trajectory made by the co-operative model deviated from the true trajectory, whereas the physics foundation model predicted 30s trajectory accurately. Fig. 3.9 shows a time history of prediction errors in the distance at 30s future made by the physics foundation model and the co-operative model. Although supportive data-driven models do not always ensure the improvement of prediction performance, it is clearly seen that the overall prediction performance of the co-operative model is much better than that of the physics foundation model.

Overall performance

Table 3.1: A summary of prediction errors in the distance in the case study.

	Mean		90% percentile	
	15s	30s	15s	30s
The physics foundation model	2.19m	7.12m	4.42m	14.7m
The co-operative model	1.09m	4.66m	2.27m	10.03m
Error reduction rate	50.2%	34.6%	48.6%	29.9%

In Fig. 3.11, the mean and 90% percentile of prediction errors in the distance over 30s prediction horizon are shown for the physics foundation model and the co-operative model, respectively. Since we have much aleatoric uncertainty in the distant future, prediction error became larger as the prediction horizon became longer. We see that the co-operative model contributed to reducing prediction error in a position significantly,

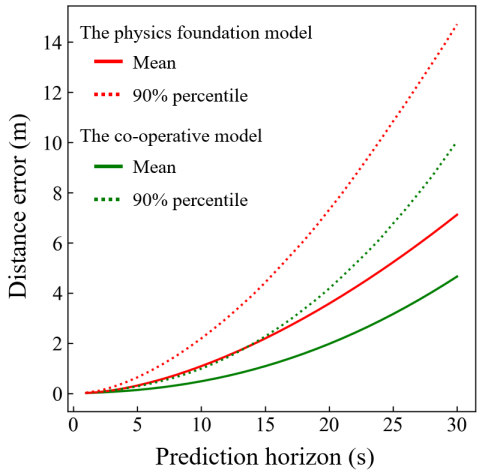


Figure 3.11: Mean and 90% percentile of prediction errors over the 30s prediction horizon.

in particular, in the distant future. It implies that the physics-data cooperation is more important for making a long trajectory prediction, which is the main interest of the situation awareness. In addition to reducing the mean prediction error in a position, the 90% percentile of prediction error in a position is notably reduced. The occurrence of large prediction errors could induce a fatal mistake in the decision making in the ship intelligence. Therefore, the improvement of 90% percentile of prediction error is one of the key contributions of the present co-operative approach. The summary of the results of the case study is shown in Tab. 3.1. At 30s future, the co-operative model reduced mean prediction error by 34.6% and 90% percentile error by 29.9% compared to that made by the physics foundation model.

3.3 Pre-training method for data-driven enhancement

This section presents research results from paper II. In other papers in this dissertation, the author assumes that the foundation model can be directly embedded as a part of the enhanced model. However, this assumption may not hold when the owners of the foundation model and those who need the enhanced model are different. For example, while the foundation model often belongs to shipbuilding companies or research institutes, shipping companies own operation data and are the ones who need to use the enhanced model. In such cases, it may be impossible to embed the foundation model into the enhanced model due to rights and technical reasons.

In this section, the author focuses on transfer learning for ML models, which transferring knowledge from source to target domains [31, 32]. Applying the idea of transfer learning into the field of ship dynamics, the author proposes the convert-mode method that enables knowledge transfer from the foundation model. The proposed method considers practical situations where the foundation model is available on the co-simulation platform, but its performance in a full-scale dataset is not satisfactory. On the other hand, there are challenges in developing a new data-driven model trained for the full-

scale dataset such as (1) the full-scale dataset is not sufficient for data-driven models to perform satisfactorily and (2) such models lack knowledge transfer from the foundation model, resulting in reduced reliability.

The proposed method creates a virtual dataset using the foundation model on the co-simulation platform. NNs are first trained with this dataset, and then they are fine-tuned using a limited full-scale dataset taken in the real-world, enabling knowledge transfer from the foundation model in an implicit manner.

3.3.1 Methodology

Pre-trianing of a trajectory prediction

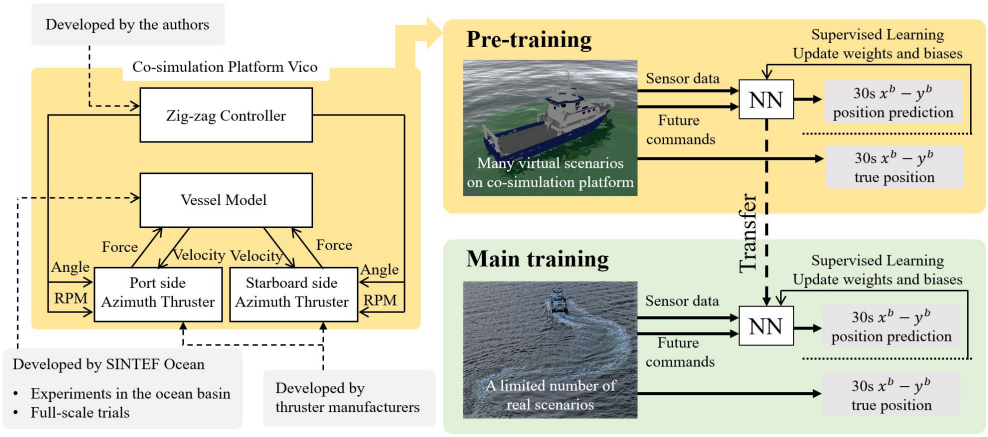


Figure 3.12: The framework of co-simulation-based development of a ship trajectory predictor.

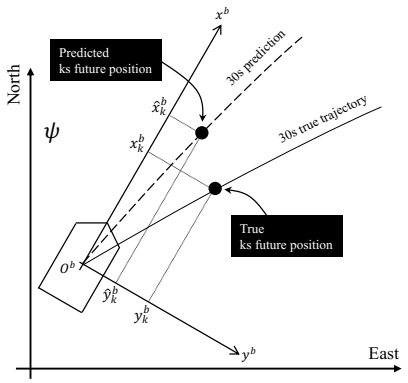


Figure 3.13: The diagram of the definition of the North-East-Down frame and the body-fixed frame when making prediction.

If we mix up the augmented and real-world data in one training phase, it may

induce a problem of how to balance those two datasets for better prediction accuracy since the fidelity of virtual simulation might not be satisfactory in most cases of ship trajectory prediction. Therefore, the main training using real-world data is conducted after the pre-training using simulation data is completed. The predictor in this study makes the trajectory prediction for 30s. Fig. 3.12 shows the overview of the present framework through co-simulation-based pre-training. In a pre-training phase shown in the right top panel in Fig. 3.12, virtual scenarios $\mathcal{D}_{\text{pre}}$ that mimic a target operation (e.g., zig-zag maneuvers in the case study) are generated in Vico. Fig. 3.13 shows the definition of the North-East-Down (NED) frame and the body-fixed $x^b - y^b - z^b$ frame when making prediction. x_k^b and y_k^b represent the true ks future x^b and y^b positions in the body-fixed frame when making prediction. ψ is the heading of the vessel to North when making prediction. $\hat{x}_k^b$ and $\hat{y}_k^b$ represent the predicted ks future x^b and y^b positions in the body-fixed frame when making prediction. An FNN f_v is pre-trained using $\mathcal{D}_{\text{pre}}$. f_v produces 30s future trajectory prediction $[\hat{x}_1^b, ..., \hat{x}_{30}^b, \hat{y}_1^b, ..., \hat{y}_{30}^b] = f_v(\nu_0, \boldsymbol{n}, \boldsymbol{\delta})$ where ν_0 is the velocity vector when making prediction, $\boldsymbol{n} = [n_0, .., n_k, ..., n_{29}]$ is the vector of thruster revolution, n_0 is the thruster revolution when making prediction, n_k is the assumption of the thruster revolution at ks future, $\boldsymbol{\delta} = [\delta_0, .., \delta_k, ..., \delta_{29}]$ is the vector of thruster revolution, δ_0 is the thruster angle when making prediction, and δ_k is the assumption of the thruster angle at ks future. In the application of this study, the prediction $[\hat{x}_1^b, ..., \hat{x}_{30}^b, \hat{y}_1^b, ..., \hat{y}_{30}^b]$ is used for evaluating the decision making $\boldsymbol{n}$ and $\boldsymbol{\delta}$. Therefore, one should note that $\boldsymbol{n}$ and $\boldsymbol{\delta}$ are given by a controller. As illustrated in the right bottom panel in Fig. 3.12, the pre-trained NN is transferred to the main training phase. In the main training phase shown in the bottom panel in Fig. 3.12, the weights and biases of f_v are finetuned using a limited number of real-world data $\mathcal{D}_{\text{train}}$.

An FNN-based predictor

In this study, we use a fully-connected FNN (= MLP) as an architecture of a predictor as it is one of the simplest and well-known ML models that is widely used in the context of transfer learning. The FNN-based predictor consists of the input layer, hidden layers, and output layer. The activation function is the hyperbolic tangent function for the hidden layers and the linear function for the output layer. ν_0, $\boldsymbol{n}$, and $\boldsymbol{\delta}$ are selected as input features through feature selection as explained hereinafter. The output of the predictor is a vector $[\hat{x}_1^b, ..., \hat{x}_{30}^b, \hat{y}_1^b, ..., \hat{y}_{30}^b]$ with a length of 60. The weights and biases of the FNN are updated so that it minimizes the Mean Squared Error (MSE) metric L between the true and predicted position vectors:

$$L = \frac{1}{H} \sum_{k=1}^{H} (\hat{x}_k^b - x_k^b)^2 + (\hat{y}_k^b - y_k^b)^2 \tag{3.5}$$

Input features are standardized with their mean and standard deviation in a training dataset in a pre-training phase.

$[x_1^b, ..., x_{30}^b, y_1^b, ..., y_{30}^b] = f(P, \nu_0, \boldsymbol{n}, \boldsymbol{\delta}) + w$ in theory of ship dynamics where P is a set of hydrodynamic and inertial parameters and w is environmental disturbances caused by wind, wave, and ocean current. In this study, we introduce following assumptions.

- A predictor is trained for a specific loading condition of a specific ship. Therefore, P is ruled out from input features of the NN.

- In most cases, a ship has no accurate measurement of waves and currents. Therefore, environmental forces due to wave and current are not modeled in the predictor.
- In $\mathcal{D}_{\text{target}}$, the effect of wind on the vessel motion is marginal. For the sake of simplicity of the validation study, it is not included in the input features of the NN.

These assumptions yield the predictor $[\hat{x}_1, ..., \hat{x}_{30}, \hat{y}_1, ..., \hat{y}_{30}] = f(\nu_0, \boldsymbol{n}, \boldsymbol{\delta})$. Hyperparameter tuning was conducted by using hyperparameter tuning framework optuna [48], of which details are found in B.

3.3.2 Validation experiment

Pre-training

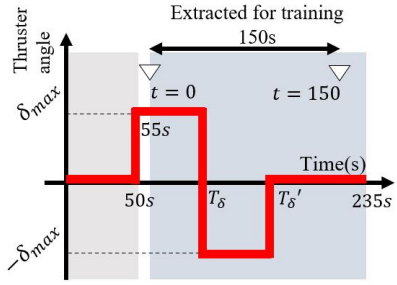


Figure 3.14: Pre-defined time series of thruster angle given by a zig-zag controller.

In a pre-training phase, $n_{\text{pre}} = 300$ virtual scenarios are generated by the virtual R/V Gunnerus in Vico. These two thrusters receive the same commands of thruster angle and revolution from a zig-zag controller. The wave, wind, and ocean current are not applied to the virtual vessel. An example of the pre-defined time series of thruster angle is shown in Fig. 3.14. Each scenario is a 235s time series. Thruster angle and revolution are set to zero before 50s. The vessel state is reset to the initial state at 50s. In order to avoid having impact load due to the reset, a $T_{\text{pre}} = 150$s time series from 55s ($t = 0$) to 205s ($t = 150$) is saved in 1Hz for the experiment with its 30s future true positions $[x^b_1, ..., x^b_{30}, y^b_1, ..., y^b_{30}]$ and corresponding controller commands at each time step. Thruster angle is $\delta_{\max}$ until $t = T_\delta$; then, it is changed to $-\delta_{\max}$ until $t = T'_\delta = 2T_\delta$ with the maximum change rate. At $t = T'_\delta$, it is turned back to zero with the maximum change rate. Thruster revolution is set to $n_{\max}$ from 50s to 235s. Each scenario is parameterized by a vector of parameters $S = [\delta_{\max}, T_\delta, n_{\max}, u_{t=0}]$ where $\delta_{\max} \in [-35°, 35°]$, $T_\delta \in [t = 50s, t = 75s]$, $n_{\max} \sim \mathcal{N}(\mu = 130\text{RPM}, \sigma = 10\text{RPM})$ and the initial surge velocity $u_{t=0} \sim \mathcal{N}(\mu = 4.0\text{m/s}, \sigma = 1.0\text{m/s})$ are randomly given to each scenario. $\mathcal{N}(\mu, \sigma)$ indicates the Gaussian distribution with the mean value μ and the standard deviation σ. The probability distribution of parameters of S can be assumed based on the general understanding of a target operation $\mathcal{D}_{\text{target}}$, however, the discrepancy of the probability distribution of input features in $\mathcal{D}_{pre}$ and $\mathcal{D}_{target}$ is inevitable. The initial North and East positions, heading, sway velocity, and yaw velocity are set to zero.

$n_{\text{pre}} = 300$ scenarios are divided into $n_{\text{train,pre}} = 192$ scenarios in the training dataset, $n_{\text{val,pre}} = 48$ scenarios in the validation dataset, and $n_{\text{test,virtual}} = 60$ scenarios in the test dataset. The test dataset is used only for checking the performance of the pre-trained predictor in the pre-training phase as shown in Fig. 3.15. The NN is trained only by using scenarios in the training dataset. To avoid overfitting the training dataset, the prediction performance of the trained NN in the validation dataset is monitored during the training.

Main training

The zig-zag maneuvers experiments of the R/V Gunnerus are conducted in November 2019 in Trondheim, Norway. Its port-side and starboard-side azimuth thrusters move simultaneously with the same commands of the thruster angle and revolution. Its tunnel thruster is turned off during the experiments. The experiment is a 1600s time history. We split this time history into $n_{\text{target}} = 16$ operations of which length is 100s. As we need 30s future positions and commands at each time step in the operation for the training and evaluation purposes, first $T_{\text{target}} = 70$s of each operation is saved as one operation with 30s true future positions $[x_1^b, ..., x_{30}^b, y_1^b, ..., y_{30}^b]$ at each time step. Equation (4.1) converts positions in the NED frame into $[x_1^b, ..., x_{30}^b, y_1^b, ..., y_{30}^b]$ in the body-fixed frame:

$$\begin{pmatrix} x_k^b \\ y_k^b \end{pmatrix} = \begin{pmatrix} \cos\psi & \sin\psi \\ -\sin\psi & \cos\psi \end{pmatrix} \begin{pmatrix} N_k - N_0 \\ E_k - E_0 \end{pmatrix} \tag{3.6}$$

where N_k and E_k are the true ks future North and East positions in the NED frame. N_0 and E_0 are North and East positions when making prediction in the NED frame. At each time step, future command assumptions $\boldsymbol{n}$ and $\boldsymbol{\delta}$ are given by the dataset as we examine the prediction performance provided that they are assumed by a controller. As we assume limited real-world data of a target operation are available, we use only $n_{\text{trainval,target}} = 12$ operations in the main training and keep the other $n_{\text{test,target}} = 4$ operations, that are used only for the evaluation of the prediction performance, untouched in the training process. $n_{\text{trainval,target}} = 12$ operations are divided into $n_{\text{train,target}} = 9$ operations in the training dataset and $n_{\text{val,target}} = 3$ operations in the validation dataset. The NN is trained only by using the training dataset and its performance in the validation dataset is monitored using the validation dataset. If the validation loss does not improve over $n_e = 200$ epochs, the training is automatically stopped; then, the best model is loaded as explained in the previous subsection. By switching the validation dataset four times, four independent NNs are trained (cross-validation). The final prediction to the untouched test dataset is the average of predictions made by these four NNs. In order to examine the contribution of the pre-training, three different strategies of training are investigated as follows.

(A) without pre-training.

The training of this predictor is conducted without a pre-training phase. Virtual scenarios generated in Vico are kept untouched and only limited real-world data of a target operation $n_{\text{train,target}}$ is used in the training. It provides a baseline of the comparison study.

(B) without finetuning.

This predictor is pre-trained with $n_{\text{train,pre}}$ virtual scenarios, however, the main training is not performed. The prediction performance of this predictor in $\mathcal{D}_{\text{target}}$ reveals the effect of the discrepancy between $\mathcal{D}_{\text{pre}}$ and $\mathcal{D}_{\text{target}}$.

(C) present study.

This is a predictor that is trained in the manner of the present framework. The main training of this predictor is carried out with real-world data $n_{\text{train,target}}$ after the pre-training with virtual scenarios $n_{\text{train,pre}}$.

Results

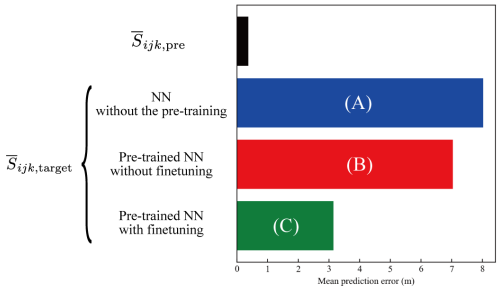


Figure 3.15: Mean prediction errors in the test dataset.

Through the optuna hyperparameter optimization in the pre-training phase, $nlayer = 1$ and $midunits = 420$ are selected. Apart from the pre-training phase, (A) without pre-training is trained only by using real-world data $n_{\text{trainval, target}}$. $nlayer = 1$ and $midunits = 300$ are selected through optuna hyperparameter optimization. After the pre-training phase, the main training is conducted. With a set of optimized hyperparameters,

Fig. 3.15 shows the mean prediction error in the test dataset in the pre-training phase $\overline{S}_{ijk,\text{pre}}$ and that in the main training phase $\overline{S}_{ijk,\text{target}}$. By comparing the bars in the top and the third from the top in Fig. 3.15, one can see that the prediction performance of the pre-trained NN deteriorates much in $\mathcal{D}_{\text{target}}$ if it is not finetuned in the main training phase due to the difference between $\mathcal{D}_{\text{pre}}$ and $\mathcal{D}_{\text{target}}$. The bar (A) in the second from the top in Fig. 3.15 reveals that the training without the pre-training phase produces the largest prediction error in (A), (B), and (C). (C) trained in the present framework with pre-training and main training phases reduces prediction error notably; by 60.8% compared to (A) without pre-training.

The left panel of Fig. 3.16 shows snapshots of 30s prediction at $t = 0$s, $t = 30$s, and $t = 60$s of one operation in the test dataset of real-world data in the main training phase. The right panel of Fig. 3.16 shows time histories of vessel state and commands of the operation. (A) the NN without the pre-training phase deviates significantly from the true trajectories at $t = 0$s and $t = 30$s. (B) the NN without fine-tuning after the pre-training phase succeeded at capturing the trend of the 30s true trajectory at $t = 0$s, $t = 30$s, and $t = 60$s in the short prediction horizon, however, it ends up with the

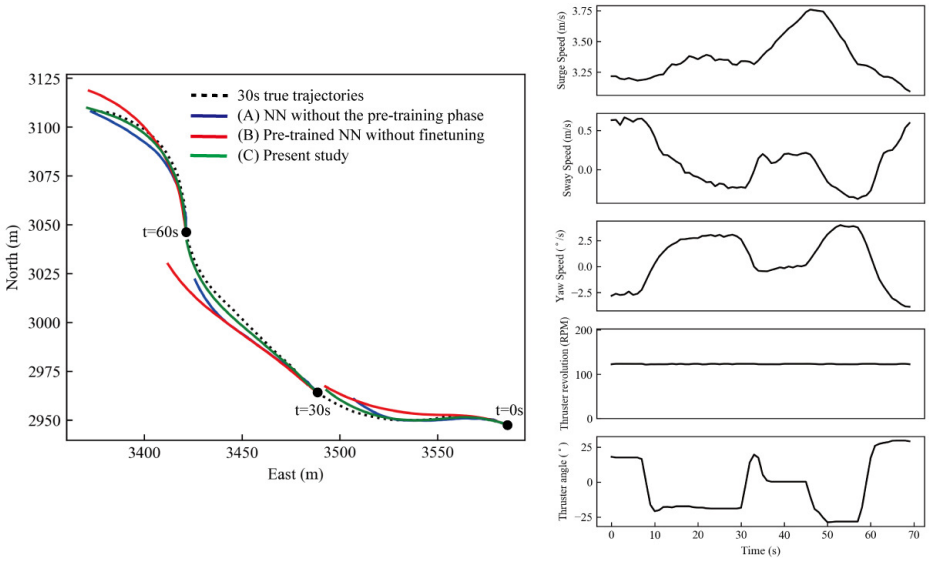


Figure 3.16: (left) Snapshots of 30s prediction at $t = 0$s, $t = 30$s, and $t = 60s$ of one scenario in the test dataset (right) Time histories of vessel state and commands.

large prediction error in the distant prediction horizon. (C) present study with the pre-training and main training phases traces the true trajectories more accurately than (A) and (B) at $t = 0$s, $t = 30$s, and $t = 60$s.

3.4 Enhancement to the full-scale DP operation

This section presents research results from paper VI. So far, the author's discussion was directed towards maneuvering operations. In contrast, this chapter explores a new method for the full-scale DP operations, where ships maintain her positions and heading by using thrusters. For ensuring operational safety, it is important to evaluate the risk of deck operations as ships are exposed to waves. The deck operations must be immediately stopped if a large wave-frequency motion is expected in the heave, pitch, and roll directions. Wave radars have been known as an option to directly measure waves surrounding a ship from onboard. It is believed that it plays a key role in a future framework of onboard decision support [57]. However, installing wave radars is usually an expensive solution. Thereby, wave-frequency motion prediction without using wave radars has been dominant [58–61]. In recent years, virtual wave radars have emerged, which are virtually installed on existing navigational radars. It makes wave radars more accessible. In fact, research has been highly limited in the field of ship motion during DP operations by using wave radars [62–66]. They conducted limited experiments using carefully tuned physics-based models. But, in practice, it is not easy to develop such models. On the contrary, in [67], neural networks were built such that they map a wave-radar observation into ship motion instead of building physics-based models, validating their work with simulation experiments. However, as discussed, pure

black-box models fully relying on data would be hardly accepted due to hurdles in training efficiency, reliability, and interpretability. Instead, this section first develops a simple physics-based model without any extensive tuning, and its data-driven mapping is introduced to improve the performance. By using a linear regression model as a mapping model, the performance of the foundation model is rapidly calibrated without losing its interpretability too much. This section employs 104 samples taken from one-year operation history of the R/V Gunnerus, which are mostly samples with small oscillation. To mitigate the negative impact of this imbalance in the dataset, this section introduces a method to improve the balance.

3.4.1 Methodology

Overview

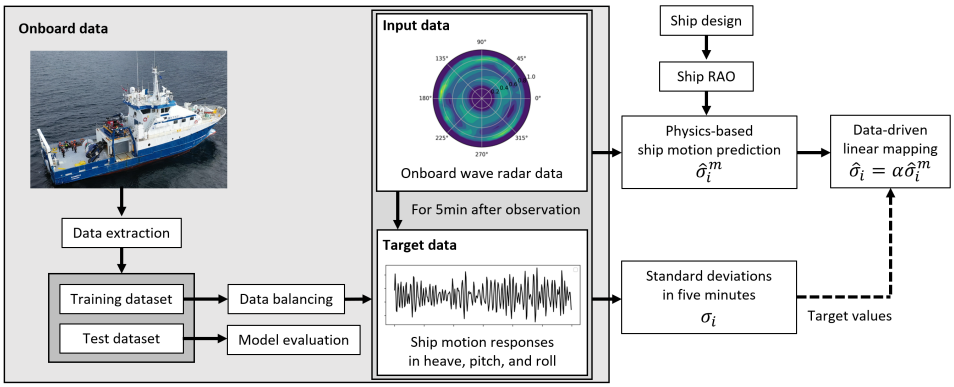


Figure 3.17: A schematic overview of the proposed approach.

A schematic overview of the proposed approach is shown in Fig. 3.17. A dataset is constructed by collecting measurements from the onboard wave radar and ship motion sensors. Taking clutter images of the sea surface, the wave radar outputs a two-dimensional wave spectrum, which is a matrix of wave energy for different wave frequencies and directions. Given that this study employs a commercial virtual wave-radar system[2], this section does not explain their technology in detail. In this study, a target of prediction is a standard deviation of ship motions σ_i, which has been used for representing how harsh ship motion is [68]. This study assumes wind and current forces have marginal impacts on this value. First, raw data are pre-processed. In this step, from long-history data, we extract samples composed of two-dimensional wave spectrum and ship motion, which satisfy criteria for data extraction. Extracted samples are grouped into training and test datasets. The test dataset is used not for model development but only for model evaluation. Subsequently, data balancing is performed to remedy a negative impact of having a poorly-balanced training dataset on the prediction performance. By using a physics-based ship linear response model, standard deviations of heave, pitch, and roll motions $\hat{\sigma}_i^m$ are predicted. A suffix i in variables represents the

[2]Miros, https://miros-group.com/

variable is defined for the ith DoF of the ship motion; namely, $i = 3$, 4, and 5 are for the heave, pitch, and roll motions, respectively. In this study, such physics-based predictions provide a baseline estimation. In the case study, true values and physics-based predictions showed a qualitative agreement, however, they had room to be calibrated. Physics-based prediction is an open-loop estimation, thereby, it accumulates errors coming from measurements, a physics model, and nonlinear ship dynamics. It takes much effort to identify multiple sources of errors and calibrate all the components properly. Moreover, it seems difficult to interfere with a data-processing system of commercial wave radars with upcoming data. Thereby, in this study, we develop a simple linear-mapping function of physics-based prediction into true values to calibrate such errors since it was found to be effective in the training dataset in the case study. Target values σ_i are given by history ship motion in the collected dataset. Once the linear function is estimated, it makes physics-data cooperative prediction for a new sample in the test dataset.

Data extraction

A raw onboard dataset gives a long time history of ship motion and two-dimensional wave spectrum taken by a virtual wave radar. The radar was not always used for the purpose of monitoring waves but for navigation, thereby, we firstly make combinations of the wave-radar observation and five-minute time series of ship motions right after the observation. Hereinafter, we call these combination samples. We remove samples of which time series have an overlap with that of the previous sample. As we focus on making ship motion prediction during DP operations, samples are further removed if the ship maneuver is not stationary. Samples with a warning of poor accuracy of wave-radar observation are also removed in this step. The remaining samples are eligible for being involved in modeling. Samples are grouped into the training and test datasets. The training dataset is used for model development. The test dataset is used only for model evaluation. When dividing samples into two datasets, they are not shuffled.

Data balancing

There exist different approaches to deal with the imbalance in a dataset. In particular, undersampling has been offering a great performance only by removing some samples from the original dataset [69]. Undersampling technique draws a line to group samples in the original dataset into rare and nominal groups. In this study, the 90% percentile of target values in the training dataset is the line dividing two groups. To balance the number of samples in two groups, undersampling technique randomly picks out samples from samples in the nominal group such that the size of the selected samples is equal to that of the rare group. Thus, undersampling improves prediction performance for samples in the rare group. It was implemented in the *resreg* package [70] in Python in this study.

Physics-based prediction

Based on the measured two-directional wave spectrum, $\hat{\sigma}_i^m$ is predicted based on the understanding of ship dynamics. Assuming the linear relationship between waves and ship motions, the wave buoy analogy method is popular [46]. Note that the encounter wave frequency is assumed to be identical to the wave frequency since we focus on DP

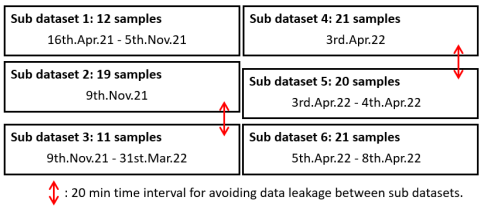


Figure 3.18: Sub datasets for the case study.

operations. In the wave buoy analogy, the relationship between the directional wave spectrum and ship motion specturm is described as (2.12). In this study, $S_w(\omega, \theta)$ measured by the wave radar is a discretized 36×32 matrix where it has 36 different wave directions with 10° interval and $N_\omega = 32$ different wave frequencies with 0.01Hz interval from 0Hz to 0.32Hz. Corresponding discretized RAO_i is calculated through a hydrodynamic workbench ShipX based on the ship's specifications and geometries. It yields a vector of energy spectrum of ship motion $S_i(\omega)$ in the discretized form. Its zeroth-order moment of m_i is:

$$m_i = \Delta\omega \sum_{k=1}^{N_\omega} S_i(\omega) \tag{3.7}$$

Then, $\hat{\sigma}_i^m = \sqrt{m_i}$ is derived for the heave, pitch, and roll motions.

Ship motion

The target of the proposed ship motion predictor is the standard deviation of the heave, pitch, and roll motions. This prediction provides stochastic insights of future ship motions for us so that we can make a decision if offshore operations can be conducted with such a sea state during DP operations. In the model training phase, these target values are given as:

$$\sigma_i = \sqrt{\frac{1}{N_t} \sum_{t=1}^{N_t} (\eta_i^t - \overline{\boldsymbol{\eta}}_i)} \tag{3.8}$$

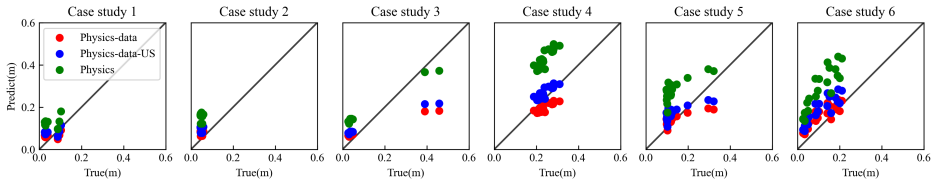
where $\boldsymbol{\eta}_i = [\eta_i^1, ..., \eta_i^{N_t}]$ is the time history of the ship motion in five minutes after the wave-radar observation, $\overline{\boldsymbol{\eta}}^i$ is its average, and N_t is the length of the time history.

Linear mapping of physics-based prediction

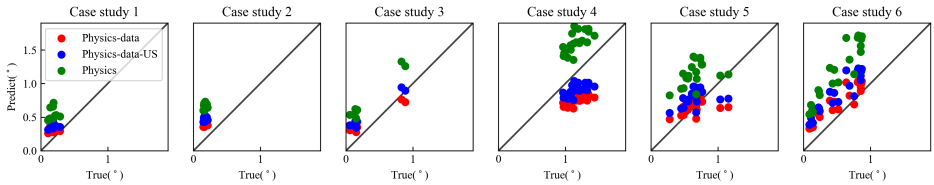
$\hat{\sigma}_i^m = \sqrt{m_i}$ are mapped into the physics-data cooperative prediction by using a linear regression function:

$$\hat{\sigma}_i = \alpha \hat{\sigma}_i^m \tag{3.9}$$

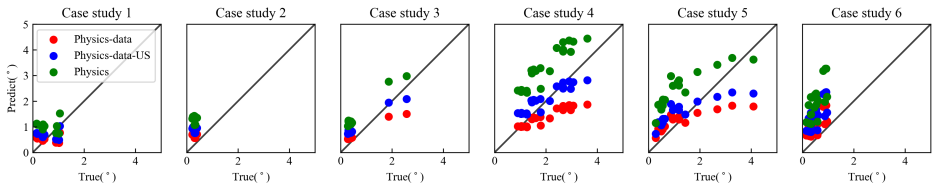
α is a coefficient of this linear-mapping function. $\hat{\sigma}_i^m$ becomes zero if the wave energy observed by the wave radar is zero. Thereby, the zero intercept of the linear mapping is estimated.



(a) Heave



(b) Pitch



(c) Roll

Figure 3.19: True versus predicted values in the (a) heave, (b) pitch, and (c) roll directions in the test datasets for all case studies. (green: Physics) predictions only with the physics-based model. (red: Physics-data) predictions with the physics-based model and its data-driven mapping by using the original training datasets. (blue: Physics-data-US) predictions with the physics-based model and its data-driven mapping by using the UnderSampled (US) training datasets.

3.4.2 Validation experiment

Data collection

Data collections were carried out from January 2021 to May 2022 by 33.9m-length RV Gunnerus. During data collection, onboard measurement of two-dimensional wave spectrum was provided by miros *Wavex* radar system [71]. *Wavex* is a commercial wave radar system virtually installed on the conventional X-band navigation radar. In the case study, it was installed on a Furuno 2xx7 series X-band navigation radar. It enables present work to be easily applied to ships with smaller costs without physically installing a new radar. *Wavex* provides two-directional spectrum based on sea clutter images taken by an X-band radar. It is connected to ship motion sensors and the impact of ship motions on sea clutter images is automatically compensated for.

Data extraction

In the collected dataset, only samples that satisfy the following criteria were extracted for the case study.

- **Data quality**: The quality of wave-radar measurements can be degraded due to factors such as wind drops and heavy precipitations. *Wavex* is equipped with automatic data quality control. Samples with unacceptable quality were removed.
- **Surge speed limit**: Samples with surge speed > 2.0 knots were removed since maneuvering operations are not the scope of this case study.
- **Steady heading**: If the heading changes over 15° within five minutes after the wave-radar observation, the samples were removed.
- **Heave motion**: The scope of the case study is the ship motion under wave excitation. If the standard deviation of the five-minute time history of the heave displacement was smaller than 0.025m, the samples were removed.

These criteria yielded 104 samples without overlaps. They were taken from diverse locations mainly off the coast of Trondheim and Ålesund. As shown in Fig. 3.18, 104 samples were grouped into six sub datasets for the case study. To avoid data leakage between sub datasets, samples in different subsets were taken on different dates or 20-min time intervals were taken between subsets.

In this section, six case studies from Case Study (CS) 1 to CS6 were conducted. For the CSk, the sub dataset k was employed as a test dataset used only for the model evaluation. The other five sub datasets were used for developing linear mapping functions.

Results

In Fig. 3.19, scatter plots of true and predicted ship motions in the test dataset in heave, pitch, and roll directions for six case studies are shown. Green, red, and blue dots represent results made by the physics-based response model, its linear mapping with original training datasets, and its linear mapping with under-sampled datasets. The diagonal lines in the figures show true lines where predicted values are equal to true values. If a dot is above the line, prediction is larger than the true value, and vice versa.

In the heave direction, the physics-based model showed a qualitative agreement with true values in all case studies, however, it was found to be mostly larger than true values. Prediction by the physics-based model is open-loop, thereby, a quantitative agreement was hardly accomplished. On the other hand, with a linear mapping trained with a corresponding training dataset, such error was notably reduced for most samples. In the CS3, the linear mapping worsened the prediction performance of the physics-based model for two samples with motion $\sigma_3 > 0.3$. It is seemingly because the training dataset of the CS3 did not have samples with such large motions. In the pitch direction, the same trend as shown for the heave direction is seen. For both heave and pitch motions, a positive impact of undersampling the training datasets was found to be marginal. By removing samples with small motions from the training datasets, prediction performance for large motions was slightly improved in return for having a slightly worse performance for small motions.

For the roll motion, it seems that the prediction performance of the physics-based model was rather limited although it captured a rough trend of true values. Linear mappings with the original training datasets notably reduced such error only for small motions. For large motion, with the original training datasets, linear mappings of the physics-based predictions underestimated the motion. It is because the original datasets are dominated by small motions, thereby, linear mapping did not efficiently learn the trend for large motions. By undersampling the training dataset, the prediction performance of linear mappings were found to be significantly improved for large motions as shown in the CS3, 4, and 5.

3.5 Chapter summary

This chapter showcased techniques for enhancing the performance of the foundation model using full-scale data taken under environmental disturbances. Three case studies were presented on maneuvering motion, and one case study was on the seakeeping motion. In these case studies, the foundation model captured the basic principles of the ship's motion and presented decent qualitative performance. However, there still existed much room for improvement in light of quantitative accuracy when it was applied to the provided datasets. The foundation model highly respects a physics consistency with experts' good understanding, thereby, building up a new branch to further lifting its performance is necessary for being benefited from the full-scale dataset. The key aspects covered in this chapter include:

- A method for correcting multiple-step-ahead prediction of the foundation model in a one-shot manner by using NNs. In particular, this chapter addressed how to handle long vectors of future command assumptions effectively in such a model. The presented method allows for learning how to improve the performance of the foundation model from the dataset while separating the contributions of the foundation model and its data-driven enhancement.

- A method for improving the performance of the foundation model closed to a layer of its basic mechanism instead of correcting the geometry of predicted trajectories. This chapter targets docking operations, where the foundation model does not satisfactorily perform due to high complexity and non-linearity. The present technique corrects the structural incompleteness of the foundation model by using a full-scale dataset.

- A method for pre-training NNs using the foundation models on simulators. This method assists in obtaining accurate models from a limited full-scale dataset by using the foundation model more implicitly. It is practical in the industry since it is not always possible to embed the foundation model directory in the system, especially for shipping companies that may not have the right to do so.

- A method for improving the accuracy of the foundation model for complex systems using full-scale monitoring data during DP operations. Ships are comprehensive systems, and the foundation model may not meet the required accuracy due to uncertainty coming from various sub-systems. In such cases, as presented, it would be an option to deliver a rapid calibration to the performance of the foundation model by using a highly interpretable data-driven model.

Hence, there is no universally best way to realize the DE framework. It is important to flexibly choose the appropriate enabler depending on pragmatic constraints and the aspects to be emphasized. In this chapter, rather than comprehensively trying out different approaches, we demonstrated the usefulness of the DE framework by targeting on priority problem settings.

4

Case study: Investigating the role of foundation models and data in enhanced performance

This section presents research results related to investigations into how the foundation model and data benefit enhanced performance, covering the contents from paper IV. In the DE framework, the foundation model and dataset are the two pillars. The problem awareness in this dissertation is the fact that we do not always have a highly-accurate foundation model and a sufficient amount of dataset. As discussed earlier, it is difficult to prepare a foundation model with satisfactory performance for the target task due to a high cost, time, and expertise. If the DE framework allows us to compromise the accuracy of the foundation model, it would be good news for the industry. On the other hand, data never cover possible operating conditions comprehensively. It is necessary to understand how the DE framework can overcome this problem. These two elements may be related to each other.

Such a key question in the DE framework has never been addressed in previous works, to the best of the author's knowledge. The DE framework can be realized in various ways as the author presented. Moreover, there are countless combinations of possible foundation models and datasets. Thereby, the author believes it would be difficult to provide a theoretical or universal answer to this research question. Rather, in this chapter, we attempt to experimentally provide one answer to this question by focusing on the most prioritized enabler of the DE framework with pragmatic problem settings.

4.1 Experimental investigation

In this section, the experimental investigations from paper IV are presented. Note that in this section, the author uses terminologies that correspond to paper IV. Therefore, different terms may be used from those used in Chapter 2 as:

- The physics-based model corresponds to the foundation model in Chapter 2.
- The data-driven compensator corresponds to the data-driven branches in Chapter 2.
- The cooperative model corresponds to the enhanced model in Chapter 2.

4.1.1 Cooperative ship model

In this study, we employ a geometry-based cooperative model, that makes a data-driven compensation for multiple-step-ahead position errors made by the foundation model. This model is an extension of the cooperative model presented in paper I.

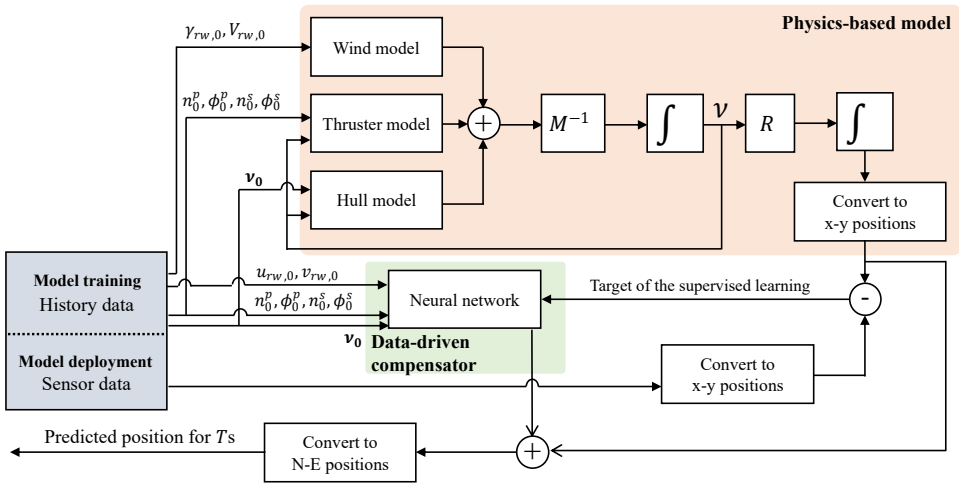


Figure 4.1: An overview of the cooperative model. Sensor data (current ship's states, thruster command values, and wind information) are given to the physics-based model and data-driven compensator. The physics-based model is a 3DOF maneuvering model outputting a trajectory prediction. The data-driven compensator, of which input-output relationship is shown in (4.3), compensates multiple-step-ahead position errors made by the physics-based model.

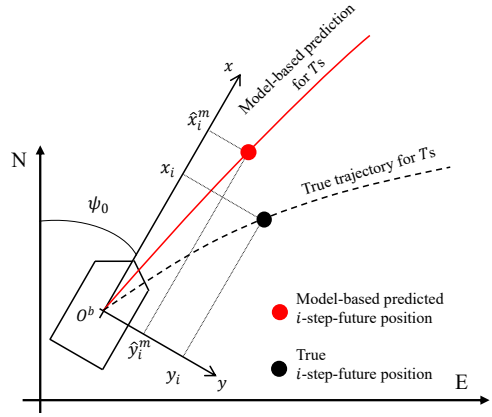


Figure 4.2: A schematic relationship between the body-fixed coordinate when making a prediction and the inertial coordinate. The data-driven compensator compensates for position errors made by the physics-based model in the body-fixed coordinate when making a prediction.

The focus of this section is to investigate the benefits of the foundation model and dataset to the cooperative performance. Thereby, it should be noted that a simpler cooperative model is introduced in this section:

- The robustness of the cooperative model against environmental disturbances is not the main theme of this section. A plain MLP is used for the data-driven compensator instead of an LSTM used in paper I.
- Unlike paper I, constant command over the prediction horizon is assumed in this study. This assumption is acceptable by carefully designing experiments in this section, thus clarifying the focus of this section.

In this subsection, the architecture of the cooperative ship model is briefly explained.

Overview

In the experiments of this study, we employed a cooperative model of ship dynamics as shown in Fig. 4.1. The cooperative model is composed of a physics-based model highlighted in orange and a data-driven compensator highlighted in green. The physics-based model makes Ts prediction of a future trajectory based on the initial state of the ship, environmental disturbances, and commands to actuators. On the other hand, based on onboard measurement data, the data-driven compensator compensates for errors in the position made by the physics-based model, Thus, a multiple-step-ahead position prediction vector made by the physics-based model is calibrated in a data-driven manner. By adding outputs of both of them, the cooperative model makes Ts prediction of the future trajectory in a data-driven manner while having a stable and reliable model-based prediction made by the physics-based model as its foundation.

Data-driven compensator

As shown in Fig. 4.2, this study expresses trajectories in the x-y coordinate of which origin is located at the center of gravity of the ship when making a prediction. The positive directions of the x and y axes are the longitudinal and lateral directions of the ship. Thereby, future positions in the x-y and N-E coordinates are interconvertible as:

$$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} \cos\psi_0 & \sin\psi_0 \\ -\sin\psi_0 & \cos\psi_0 \end{pmatrix} \begin{pmatrix} N - N_0 \\ E - E_0 \end{pmatrix} \tag{4.1}$$

where N_0, E_0, and ψ_0 represent the north, east positions and heading when making a prediction. Hence, the model-based predicted trajectory $[\hat{N}_1^m, ..., \hat{N}_{n_T}^m, \hat{E}_1^m, ..., \hat{E}_{n_T}^m]$ in the N-E coordinate is converted to $[\hat{x}_1^m, ..., \hat{x}_{n_T}^m, \hat{y}_1^m, ..., \hat{y}_{n_T}^m]$ in the x-y coordinate to provide target vectors of the data-driven compensator.

The data-driven compensator makes prediction of $[\Delta\hat{x}_1^m, ..., \Delta\hat{x}_{n_T}^m, \Delta\hat{y}_1^m, ..., \Delta\hat{y}_{n_T}^m]$ where $\Delta\hat{x}_i^m = x_i - \hat{x}_i^m$ and $\Delta\hat{y}_i^m = y_i - \hat{y}_i^m$, respectively, based on onboard sensor measurements. x_i and y_i are the true position at the i-step future. This target vector is given not in the model deployment but in the model training.

Input features; This study selects input features based on the theory of ship dynamics that future trajectories are determined by inertial and hydrodynamic parameters of the ship, the initial state of the ship, commands to thrusters, and environmental

disturbances. As we develop data-driven compensators for the specific loading condition of the specific ship, the inertial and hydrodynamic parameters are assumed to be constant. Wave and ocean current data are not included in input features as they are mostly not measured in real time. We assume commands to thrusters, the true wind velocity, and the true wind direction are kept unchanged over the prediction horizon. These assumptions yield the formulation of the data-driven compensator f_N:

$$[\Delta\hat{x}_1^m, ..., \Delta\hat{x}_{n_T}^m, \Delta\hat{y}_1^m, ..., \Delta\hat{y}_{n_T}^m] = f_N(\boldsymbol{\nu_0}, n_0^p, \delta_0^p, n_0^s, \delta_0^s, u_{rw,0}, v_{rw,0}) \tag{4.2}$$

The suffix 0 represents the values when making a prediction. The input vector is z-score normalized with the statistic values in the training dataset. The same values are applied to the normalization in the validation and test datasets. In the experiments of this study, two azimuth thrusters are manipulated with the same commands. Thereby, (4.2) is reduced to:

$$[\Delta\hat{x}_1^m, ..., \Delta\hat{x}_{n_T}^m, \Delta\hat{y}_1^m, ..., \Delta\hat{y}_{n_T}^m] = f_N(\boldsymbol{\nu_0}, n_0, \delta_0, u_{rw,0}, v_{rw,0}) \tag{4.3}$$

where $n_0 = n_0^p = n_0^s$ and $\delta_0 = \delta_0^p = \delta_0^s$.

Model training; This study employs an MLP, which uses tanh and linear functions for the hidden layer(s) and the output layer, respectively. During training, we separate some maneuvers from a training-validation dataset and keep them as a validation dataset. The validation loss is monitored to avoid overfitting the training dataset. If the validation loss does not improve over 200 epochs, the training is automatically terminated, and the best model is used for the prediction (early stopping).

Hyperparameter tuning; Hyperparameters are parameters to be fixed in advance to determine ML model's architecture and training setting. Hyperparameter tuning is important to achieve a good performance of ML models. In this study, the number of hidden layers $\in [1, 3]$, the number of units in hidden layer(s) $\in [10, 500]$, the drop-out rate in the input layer $\in [0.0, 1.0]$, the drop-out rate in hidden layer(s) $\in [0.0, 1.0]$, and the learning rate of the optimizer $\in [10^{-5}, 10^{-1}]$ are optimized. The number of trials for the parameter search is 50 as further drastic improvement of the validation loss was not found with the larger number of trials than 50. After 50 trials of the hyperparameter search, a set of hyperparameters with the best performance in the validation dataset was selected as a set of optimum hyperparameters. Hyperparameter tuning was conducted independently for having different physics-based models and dataset.

Model deployment; In the model deployment, the data-driven compensator makes prediction $[\Delta\hat{x}_1^m, ..., \Delta\hat{x}_{n_T}^m, \Delta\hat{y}_1^m, ..., \Delta\hat{y}_{n_T}^m]$ based on input vectors provided by onboard sensors. By adding the model-based predicted position vector $[\hat{x}_1^m, ..., \hat{x}_{n_T}^m, \hat{y}_1^m, ..., \hat{y}_{n_T}^m]$, the cooperative prediction yields $[\hat{x}_1, ..., \hat{x}_{n_T}, \hat{y}_1, ..., \hat{y}_{n_T}]$ where $\hat{x}_i = \Delta\hat{x}_i^m + \hat{x}_i^m$ and $\hat{y}_i = \Delta\hat{y}_i^m + \hat{y}_i^m$.

Evaluation metrics; The accuracy of the physics-based model A is evaluated with the Root Mean Squared Error (RMSE) of the geometrical similarity between true and predicted trajectories in the test dataset.

$$A = \frac{1}{S}\sum_{k=1}^{S}\sqrt{(N_k - \hat{N}_k^m)^2 + (E_k - \hat{E}_k^m)^2} \tag{4.4}$$

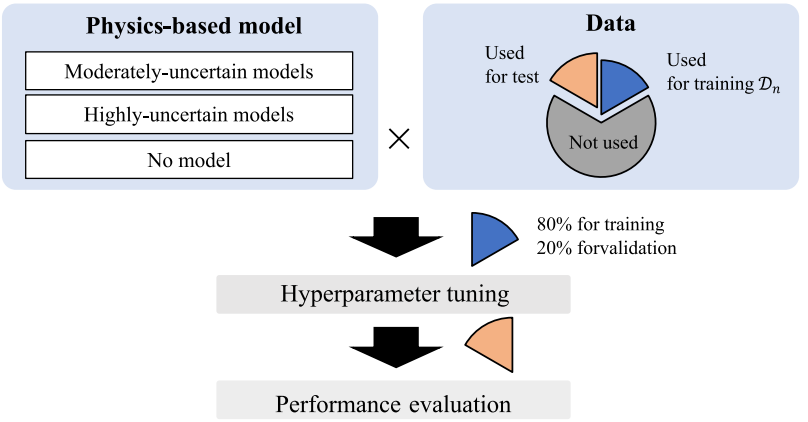


Figure 4.3: Experimental setting.

N_k and E_k represent the true north and east positions of the k-th sample. $\hat{N}_k^m$, and $\hat{E}_k^m$ are the north and east positions of the k-th sample predicted by the physics-based model. S is the number of data samples. N_D represents the number of maneuvers in the dataset used for the training. The larger the N_D is, the larger the dataset is. Hence, cooperative models are characterized by the combination (A, N_D) in this study. It should be noted that the ship dynamics is highly complex and nonlinear, thereby, a single metric A does not fully represent the characteristic of the physics-based model.

The errors made by the cooperative model H of (A, N_D) is evaluated with the RMSE of the geometrical similarity between the true and predicted trajectories made by the cooperative model:

$$H = \frac{1}{S}\sum_{k=1}^{S}\sqrt{(N_k - \hat{N}_k)^2 + (E_k - \hat{E}_k)^2} \tag{4.5}$$

where $\hat{N}_k$ and $\hat{E}_k$ are the north and east positions of the k-th sample, predicted by the cooperative model. It should be noted that A is not used for selecting a good physics-based model for better cooperative performance since it is defined in the test dataset. A is only employed for presenting the relationship between H and (A, N_D) in the test dataset to develop a better understanding of the contribution of the physics-based model and data to the cooperative performance. If $N_D = 0$, we substitute $\hat{N}_k^m$ and $\hat{E}_k^m$ for $\hat{N}_k$ and $\hat{E}_k$ in graphs since the cooperative model without using any data is regarded as the physics-based model. If the pure data-driven model is used, $\hat{N}_k$ and $\hat{E}_k$ are calculated without the help of the physics-based model.

4.1.2 Simulation experiment

Before performing full-scale experiments, we examine the contribution of the physics-based model and data to the cooperative performance in the simulation environment. The cooperative model was trained and evaluated for making $T = 30$s trajectory prediction.

Overview

The overview of simulation experiments in this study is illustrated in Fig. 4.3. 18 different physics-based models were used in the cooperative models. A dataset with 120 maneuvers was prepared in this study. 20 maneuvers in the dataset were randomly selected for the test dataset and kept untouched during training and validation of the cooperative models. By selecting N_D maneuvers from the remaining 100 maneuvers, we built the training and validation sub dataset $\mathcal{D}_{N_D}$ with the different number of maneuvers. Please note that $\mathcal{D}_{N_D=a} \subset \mathcal{D}_{N_D=b}$ if $a < b$. In this study, ten sub datasets from $\mathcal{D}_{N_D=10}$ to $\mathcal{D}_{N_D=100}$ were prepared. We trained the cooperative models with different physics-based models and sub datasets; thus examining the impact of the accuracy of the physics-based model and dataset on the cooperative performance. For the different combinations of the physics-based models and sub datasets, hyperparameter tuning was conducted independently. In $\mathcal{D}_{N_D}$, 80% of maneuvers were used for the training and the remaining 20% were used for the validation. The performance of the trained cooperative model was examined by using 20 maneuvers in the test dataset. The test dataset was always identical regardless of which sub dataset was used during training.

Dataset

A simulation dataset was generated by a virtual ship of the R/V Gunnerus on co-simulation platform Vico. 120 unique turning maneuvers were generated by randomly selecting thruster revolution $n \in [50, 200]$ Revolution Per Minute (RPM), and thruster angle $\delta \in [-50, 50]^\circ$. The ship's motion was disturbed by the constant wind and irregular waves in the simulation. The true wind direction $\beta_w \in [0, 360)^\circ$, the true wind speed $V_w \in [0, 6]$ m/s, and the global wave direction $\in [0, 360)^\circ$ are randomly chosen for each maneuver. The wave spectrum was JONSWAP spectrum [56] with 1.0m significant wave height and 5.0s significant wave period. 50s time series were saved in 1Hz with 30s future trajectory at each time step for each maneuver. Future trajectories were used only for training and evaluation purposes.

Physics-based models

In this experiment, cooperative models were trained with different physics-based models. They were developed by shifting parameters of the ground-truth model used in the simulation environment. This procedure introduced the model's uncertainty that we had in reality due to poorly identified parameters. We randomly produced 18 physics-based models with parameter uncertainty to examine the impact of the accuracy of the physics-based model on the cooperative performance. 18 models were grouped into moderately- and highly-uncertain models. It should be noted that none of them were identical to the ground-truth model used in the simulation. They produced prediction errors at different levels due to different reasons.

Moderately-uncertain models; We prepared ten physics-based models by randomly shifting parameters of the ground-truth model in $\boldsymbol{D}(\boldsymbol{\nu_r})$. They were grouped into moderately-uncertain models in this paper. The ground-truth model has 32 hydrodynamic derivatives θ_1-θ_{32} in $\boldsymbol{D}(\boldsymbol{\nu_r})$ (such as X^L_{uu}, see [72] for details). A set of disturbed parameters $\theta'_{i,j}$ the i-th hydrodynamic derivative of the j-th moderately-uncertain model was introduced as:

$$\theta'_{i,j} = \Delta_{i,j}\theta_i \tag{4.6}$$

$0.4 < \Delta_{i,j} < 1.6$ was randomly selected for the i-th hydrodynamic derivative of the j-th moderately-uncertain model. Although they made prediction errors due to poorly identified parameters, predicted trajectories they made could represent the basic characteristics of the true dynamics of the targeting ship. The mass, inertia moment, added-mass coefficients, and thruster models were kept unchanged from the ground-truth model. This situation could occur if we copy and paste hydrodynamic parameters of similar ships, we have a physics-based model adjusted to the different operational conditions, and so on.

Highly-uncertain models; We prepared another eight physics-based models by randomly shifting the mass, inertia moment, added-mass coefficients, and propeller diameter of the thruster model up to 40%, in the same procedure as (4.6), in addition to the parameter shift introduced in the moderately-uncertain models. The trajectories they predicted had very different characteristics from the true trajectories since the basic parameters of the model were shifted. This situation could occur if we copy and paste parameters of very different ships or actuator models have large uncertainty.

No model; If no model was assigned to the physics-based model, a pure data-driven model was built in the experiment. It was trained in the same manner as the cooperative model, however, the target vector was not the residual vector $[\Delta\hat{x}_1^m, ..., \Delta\hat{x}_{n_T}^m, \Delta\hat{y}_1^m, ..., \Delta\hat{y}_{n_T}^m]$ but the future position vector $[\hat{x}_1, ..., \hat{x}_{n_T}, \hat{y}_1, ..., \hat{y}_{n_T}]$ without the help of any model-based guides.

Results

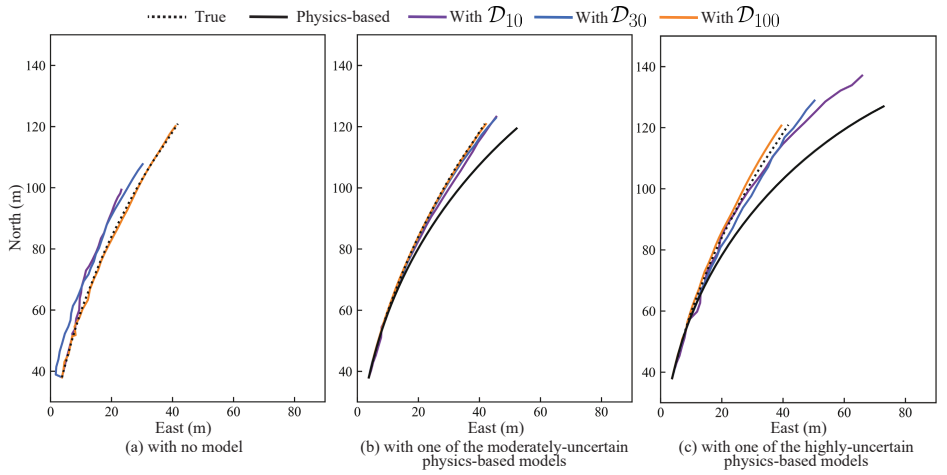


Figure 4.4: Snapshots of trajectory predictions made by (a) the pure data-driven model, (b) the cooperative model with one of the moderately-uncertain physics-based models, and (c) the cooperative model with one of the highly uncertain physics-based models with sub dataset $\mathcal{D}_{10}$, $\mathcal{D}_{30}$, and $\mathcal{D}_{100}$.

Snapshots in Fig. 4.4 show predictions with different physics-based models with different sub datasets at one of the example prediction time instance in the test dataset.

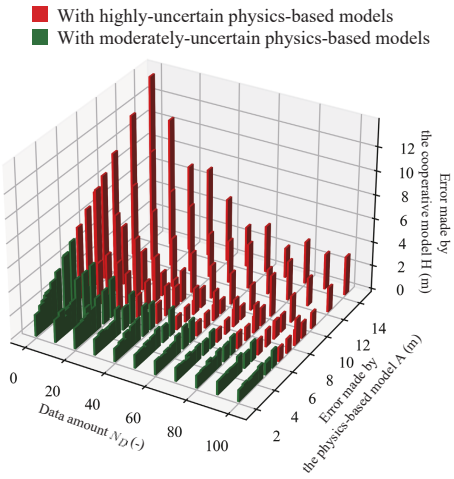


Figure 4.5: The effect of the errors of the physics-based model A and the data amount N_D on the errors made by the cooperative model H in the test dataset.

Black dotted lines show the 30s true trajectories, which are the same in the three subfigures. In Fig. 4.4 (a), predicted trajectories made by pure data-driven models trained with $\mathcal{D}_{10}$, $\mathcal{D}_{30}$, and $\mathcal{D}_{100}$ are shown. Since they were not supported by any prior knowledge of ship dynamics, it is seen that they needed a large dataset to make predictions accurately. Models trained with $\mathcal{D}_{10}$ and $\mathcal{D}_{30}$ ended up making discontinuous trajectories with less similarity to the true trajectory. In Fig. 4.4 (b), predicted trajectories made by cooperative models with one of the moderately uncertain physics-based models with $\mathcal{D}_{10}$, $\mathcal{D}_{30}$, and $\mathcal{D}_{100}$ are shown. Although physics-based models made prediction error in (b), it was rather small and captured the basic geometry of the true trajectory. It is seen that data-driven compensators compensated for such errors well only by using a small dataset $\mathcal{D}_{10}$. As the pure data-driven model with $\mathcal{D}_{10}$ failed at making an accurate prediction in (a), it implies that the physics-based model successfully supported the cooperative performance. In Fig. 4.4 (c), predicted trajectories made by cooperative models with one of the highly-uncertain physics-based models with $\mathcal{D}_{10}$, $\mathcal{D}_{30}$, and $\mathcal{D}_{100}$ are shown. In (c), a trajectory predicted by the physics-based model notably diverges from the true trajectory. The poor performance of the physics-based model was induced by its parameters with higher uncertainty. We see it deteriorated the cooperative performance with $\mathcal{D}_{10}$ and $\mathcal{D}_{30}$ significantly while the performances in (b) were very good with the same datasets. Moreover, although the large prediction error was mitigated by having a large sub dataset $\mathcal{D}_{100}$, the cooperative performance with $\mathcal{D}_{100}$ in (b) outperforms that in (c).

An overview of results is illustrated in Fig. 4.5. This figure shows the relationship between the errors made by the cooperative model H as a height of bars and its (A, N_D) as a position of bars on the bottom plane, where A denotes the errors made by the physics-based model and N_D denotes the number of maneuvers in the training dataset. For instance, the height of the bar located at the small A and large N_D on the bottom

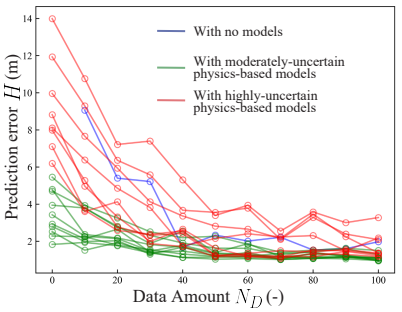


Figure 4.6: A projected 2D graph of the effect of the accuracy of the physics-based model A and the data amount N_D on the errors made by the cooperative model H in the test dataset.

plane shows the errors made by the cooperative model with a combination of such an accurate physics-based model and a large dataset for the training. Bars at $N_D = 0$ shows the original performance of the physics-based model without using any data for the training. We see that most cooperative models with highly-uncertain physics-based models made larger errors than cooperative models with moderately-uncertain physics-based models with the same amount of data for training. In addition, a trend was seen that the higher cooperative performance was achieved with a larger dataset and a more accurate physics-based model. A good cooperative performance was achieved by either having an accurate physics-based model or having a large dataset; thereby, they are complementary to each other to some extent.

Fig. 4.6 is a projected graph of Fig. 4.5 on the N_D-H plane for better visibility of absolute values of the cooperative performance. In Fig. 4.6, a trend is seen that the higher the accuracy of the physics-based model was, the higher the cooperative performance was, especially when the dataset was small. At the same time, cooperative models with a wide range of physics-based models, including some highly-uncertain models with relatively better performance, were found to outperform the pure data-driven model. It implies the possibility of the cooperative framework of building an accurate model with a compromised physics-based model and a small dataset. However, it does not mean any physics-based models are acceptable as a foundation of cooperative models. It is clearly seen that some cooperative models with highly-uncertain physics-based models ended up with poorer performance than the pure data-driven models. In such cases, physics-based models seem to not introduce prior knowledge of ship dynamics but introduce disturbances in the training. Thereby, the negative impact of having such physics-based models with the poor performance on the cooperative performance remained even if we had a large dataset. In particular, the performance of the cooperative model with the most inaccurate physics-based model fluctuated much depending on the sub dataset used for the training. It is seemingly caused by its high data dependency with a physics-based model introducing disturbance to the training.

4.1.3 Full-scale experiment

Hereinafter, we further explore the reasonable range of the physics-based model's accuracy on the cooperative performance in the real-life project by employing a small dataset of a full-scale experiment in the open sea. This full-scale experiment validates that we can build an accurate ship dynamic model in the practical project by combining a compromised physics-based model and a small dataset rather than relying on either of them.

We made $T = 15$s trajectory predictions in the full-scale experiment. In the full-scale experiment, we investigated the cooperative performances with different physics-based models and a small dataset to examine the framework building an accurate model with a compromised physics-based model and a small dataset.

Dataset

We conducted full-scale experiments in the open sea on November 21st, 2019 in Trondheim, Norway. The 33.9m-length R/V Gunnerus was employed. Under the mild weather condition, We conducted 10°/10°, 15°/15°, 20°/20°, 25°/25°, and 30°/30° zigzag maneuvers with high ($n \approx 145$RPM) and low ($n \approx 125$RPM) surge velocities. Each maneuver was saved in 1Hz and cut into 85s time series with 15s future positions at each time step. The number of sampled maneuvers was 16. A 20°/20° zigzag maneuver with the high surge velocity was kept untouched for the test dataset. This maneuver was not included in the other maneuvers in the training-validation dataset. During maneuvers, same commands were given to the two azimuth thrusters and the bow thruster was turned off. Except for the maneuver in the test dataset, 15 maneuvers were used for the training. The three-fold cross validation was conducted by using 15 maneuvers in the training and validation dataset.

Physics-based models

In the full-scale experiment, we employed two physics-based models; namely, accurate and inaccurate physics-based models to examine the impact of having different physics-based models on the cooperative performance with a real-life small dataset.

Accurate physics-based model; Before the full-scale experiment, the R/V Gunnerus was elongated from 28.9m to 33.9m. However, the corresponding ship dynamic model has not been fully developed. In this study, we employ a ship dynamic model of the 28.9m R/V Gunnerus before the elongation as a physics-based model since it well captures the dynamic behavior of the elongated R/V Gunnerus as well. It is referred to as the accurate physics-based model in this experiment. It represents an optimistic assumption that an accurate physics-based model is available in the project.

Inaccurate physics-based model; In an inaccurate physics-based model, we shifted dominant damping coefficients in addition to removing higher-order damping coefficients in $\boldsymbol{D}(\boldsymbol{\nu_r})$. The inaccurate physics-based model represents a pessimistic assumption that the available physics-based model performs poorly due to different reasons.

No model; If no physics-based model was given, as we did in the simulation experiment, a pure data-driven model was built without the help of the physics-based model.

Results

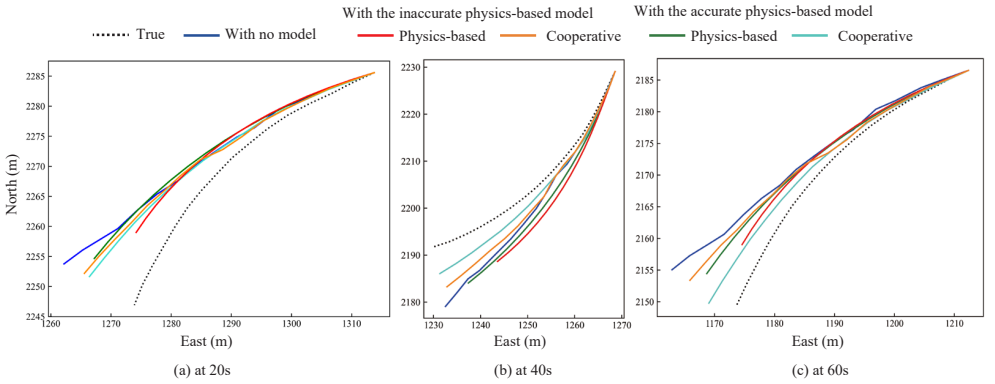


Figure 4.7: Snapshots of trajectory predictions at (a) 20s, (b) 40s, and (c) 60s of the maneuver in the test dataset.

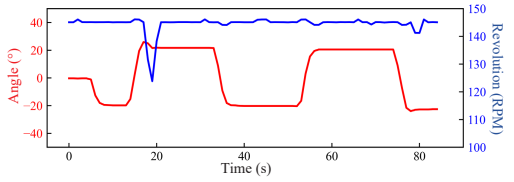


Figure 4.8: A time series of thruster angle δ and revolution n in the example maneuvers in the test dataset shown in Fig. 4.7.

In Fig. 4.7, snapshots of 15s trajectory predictions of the maneuver in the test dataset at $t = 20$s in (a), $t = 40$s in (b), and $t = 60$s in (c) are shown. Time histories of the thruster revolution and angle of this maneuver are shown in Fig. 4.8. In Fig. 4.7 (a), although the cooperative model with the accurate physics-based model made smaller errors compared to the other models, all models deviated from the true trajectory notably. In the full-scale experiment, as the dataset used for training was limited, it was seen that cooperative models did not always make a good prediction seemingly due to the lack of experience during training. On the other hand, in (b) and (c), cooperative models notably reduced prediction errors made by the corresponding physics-based models and significantly outperformed the pure data-driven model. In (b), the cooperative model with the inaccurate physics-based model performed better than the accurate physics-based model. In (c), its performance was comparable to the accurate physics-based model. Hence, although the cooperative model with the inaccurate physics-based model did not outperform that with the accurate physics-based model, the contribution of having such a compromised physics-based model was clearly discerned. This finding corresponds to the results presented in the simulation experiment.

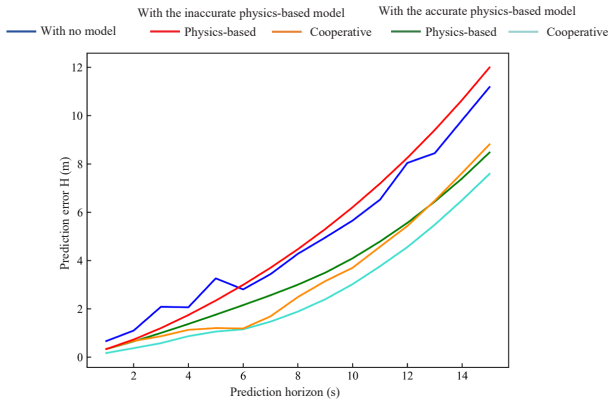


Figure 4.9: The prediction error H over the prediction horizon made by pure data-driven, pure physics-based, and cooperative models.

Fig. 4.9 shows the average prediction error H in the 15s prediction horizon at each prediction instance made by the pure data-driven, pure physics-based, and cooperative models. The prediction error becomes larger in the distant horizon as we have much uncertainty in the distant future. The pure data-driven model made larger prediction errors than the other models over the prediction horizon. It highlights the benefit of having the physics-based model in terms of prediction performance. Although the inaccurate physics-based model did not perform well in the experiment over the prediction horizon, its prediction error was well compensated for by using the small dataset. Its overall performance is comparable to that of the accurate physics-based model, which requires much time & effort to be developed. In addition, in Fig. 4.9, the discrepancy of the prediction error H made by the accurate and inaccurate physics-based model was found to be large. On the other hand, by using the small dataset, the discrepancy between corresponding cooperative performances becomes smaller. It implies the robustness of the cooperative model to the poor accuracy of the physics-based model.

4.2 Chapter summary

This chapter addressed a key research question in the DE framework, which is how the foundation model and data benefit enhanced performance. The goal is to build an enhanced model that shows high performance by combining these two pillars while compromising both of them to some extent. The theoretical investigation of this question is challenging since the DE framework can be enabled through various methods with numerous combinations of the foundation model and data. Therefore, this chapter presented one solution to this question in an experimental manner on realistic problem settings. In simulation experiments, we comprehensively investigated the impact of various combinations of foundation models and datasets of different sizes on enhanced performances. In the full-scale experiment, we demonstrated the possibility of enhanced models that use compromised foundation models and a limited amount of full-scale data.

This chapter experimentally revealed that the foundation model and data complement each other to enhanced performance. It is shown that we would have no such thing as a general ship model that serves as a foundation model of any ships. The presented results suggest that it is more important to balance these pillars rather than being overly attached to building accurate physics-based models or collecting massive amounts of data. This is a key experimental finding that demonstrates the promise of the present DE framework.

5

Case study: Knowledge transfer between similar ships

So far, the DE framework has been directed towards readily reflecting the full-scale dataset to model performance. From a different perspective, this chapter explores a method to bridge similar ships' dynamics, covering the contents from Paper V.

Recently, the system identification (SI) method has gained attention for developing physics-based models for ship maneuvering. Traditionally, it has been relying on numerical or model-scale captive model tests (CMT), which were regarded as the most reliable data sources with measurements of forces and moments. However, these experiments require advanced expertise and significant costs. Instead, the SI method statistically determines the parameters of physics-based models by directly using the full-scale ship operation data. It requires less cost and avoids scale effect on model performance. On the other hand, it also comes with drawbacks. In the SI method, the identified model can be easily affected by perturbations in the dataset. This is well-known as the parameter drift problem. To address this issue, measures such as (1) optimizing excitation signals [73] and experiment design [74], (2) simplifying model structures [75–77], and (3) using robust identification algorithms [78, 79] have been taken.

To offer one solution to this issue, this study proposes a new framework called **model refinement** in the SI approach by bridging two similar ships. The proposed technique belongs to the update mode in this dissertation. When building a maneuvering model for a new ship with its free-running data, a model built in the CMT approach for a similar ship may be available. In this study, the former ship is referred to as a source ship acting as a knowledge safeguard. The latter ship is referred to as a target ship being a target of the PI. In such a case, designers would need to prioritize the model for the similar ship into building the new model for the new ship for having consistency between models for two similar ships. Otherwise, parameters are searched globally highly depending on the collected dataset. In addition, it is hard for designers and stakeholders to have sufficient confidence in model's validity if a new model is detached from connections between models. To prioritize such a connection in modeling and bridge dynamics of similar ships, this study employs parameters of the similar ship as a l_2 constraint in the ridge regression algorithm when identifying parameters for the new ship with free-running data of the new ship. It aims at refusing to update parameters into the range where designers do not justify such an update. Parameters of the similar ship are introduced as a safeguard in the data-driven calibration for controlling the degree of contribution of data infused into such a knowledge foundation. It helps designers find a robust-and-accurate model within the safe zone. The proposed framework gives models of similar ships the value as a knowledge asset bringing such benefits into building a new model for a new ship.

5.1 Knowledge transfer between similar ships

In the traditional approach, experts involve their knowledge mostly in the design of experiments, models, and identification algorithms. It requires much cost, time, and expertise to conduct them carefully. Then, the hyperparameters of the identification algorithm are tuned. The PI is performed by using data. Experts and stakeholders at this moment judge the model's validity and check its performance for the selected maneuvers [80]. Such evaluations are rather passive. It may result in carefully investigating, improving, or redoing some processes if they are less confident in the model's validity. This iteration seems inefficient when we have confidence in the similarity between resulting models of two similar ships.

On the contrary, by building up models on top of the source parameters, the proposed framework actively employs knowledge coming from past experience for similar ships as a modeling foundation. Firstly, experts and stakeholders need to agree that it is reasonable that the target model is derived by refining the source model in light of their domain knowledge and experience. This agreement allows the source model to act as a source of modeling confidence. Then, we highly constrain parameters towards the source parameters by defining conservative ranges of hyperparameters to do so. It could occur that we do not find parameters satisfying improvement criteria with defined ranges of hyperparameters. Then, the ranges are broadened in the next iteration leading to less prioritized the source parameters and more dependency on data. Hence, the proposed framework parsimoniously finds a model by gradually expanding the tuning ranges starting from the source parameters in a parameter space.

5.1.1 Methodology

Ship maneuvering model

The non-dimensional forms of hydrodynamic forces and moments in the 3DoF Abkowitz model are descrived as:

$$\begin{aligned}\Delta f_1' =& X_u'\Delta u' + X_{uu}'\Delta u'^2 + X_{uuu}'\Delta u'^3 + X_{vv}'\Delta v'^2 + X_{rr}'\Delta r'^2 \\ &+ X_{\delta\delta}'\Delta\delta'^2 + X_{\delta\delta u}'\Delta\delta'^2\Delta u' + X_{vr}'\Delta v'\Delta r' \\ &+ X_{v\delta}'\Delta v'\Delta\delta' + X_{v\delta u}'\Delta v'\Delta\delta'\Delta u' + X_{uvv}'\Delta u'\Delta v'^2 \\ &+ X_{urr}'\Delta u'\Delta r'^2 + X_{uvr}'\Delta u'\Delta v'\Delta r' + X_{r\delta}'\Delta r'\Delta\delta' \\ &+ X_{ur\delta}'\Delta u'\Delta r'\Delta\delta' + X_0'\end{aligned} \tag{5.1}$$

$$\begin{aligned}\Delta f_2' =& Y_{0u}'\Delta u' + Y_{0uu}'\Delta u'^2 + Y_v'\Delta v' + Y_r'\Delta r' + Y_\delta'\Delta\delta' \\ &+ Y_{vvv}'\Delta v'^3 + Y_{\delta\delta\delta}'\Delta\delta'^3 + Y_{vvr}'\Delta v'^2\Delta r' \\ &+ Y_{vv\delta}'\Delta v'^2\Delta\delta' + Y_{v\delta\delta}'\Delta v'\Delta\delta'^2 + Y_{\delta u}'\Delta\delta'\Delta u' \\ &+ Y_{vu}'\Delta v'\Delta u' + Y_{ru}'\Delta r'\Delta u' + Y_{\delta uu}'\Delta\delta'\Delta u'^2 + Y_{rrr}'\Delta r'^3 \\ &+ Y_{vrr}'\Delta v'\Delta r'^2 + Y_{vuu}'\Delta v'\Delta u'^2 + Y_{ruu}'\Delta r'\Delta u'^2 \\ &+ Y_{r\delta\delta}'\Delta r'^2\Delta\delta' + Y_{rr\delta}'\Delta r'^2\Delta\delta' + Y_{rv\delta}'\Delta r'\Delta v'\Delta\delta' + Y_0'\end{aligned} \tag{5.2}$$

$$
\begin{aligned}
\Delta f_3' =& N_{0u}'\Delta u' + N_{0uu}'\Delta u'^2 + N_v'\Delta v' + N_r'\Delta r' + N_\delta'\Delta\delta' \\
&+ N_{vvv}'\Delta v'^3 + N_{\delta\delta\delta}'\Delta\delta'^3 + N_{vvr}'\Delta v'^2\Delta r' \\
&+ N_{vv\delta}'\Delta v'^2\Delta\delta' + N_{v\delta\delta}'\Delta v'\Delta\delta'^2 + N_{\delta u}'\Delta\delta'\Delta u' \\
&+ N_{vu}'\Delta v'u' + N_{ru}'\Delta r'\Delta u' + N_{\delta uu}'\Delta\delta'\Delta u'^2 + N_{rrr}'\Delta r'^3 \\
&+ N_{vrr}'\Delta v'\Delta r'^2 + N_{vuu}'\Delta v'\Delta u'^2 + N_{ruu}'\Delta r'\Delta u'^2 \\
&+ N_{r\delta\delta}'\Delta r'^2\Delta\delta' + N_{rr\delta}'\Delta r'^2\Delta\delta' + N_{rv\delta}'\Delta r'\Delta v'\Delta\delta' + N_0'
\end{aligned} \tag{5.3}
$$

where $\{X_{(\cdot)}', Y_{(\cdot)}', N_{(\cdot)}'\}$ are hydrodynamic derivatives to be identified. $\Delta u'$, $\Delta v'$, and $\Delta r'$ are non-dimensional surge, sway, and yaw velocities. Five zero-frequency added-mass derivatives $X_{\dot{u}}'$, $Y_{\dot{v}}'$, $Y_{\dot{r}}'$, $N_{\dot{v}}'$, and $N_{\dot{r}}'$ are usually calculated precisely by using semi-empirical formulas or a strip-theory calculation, thereby, they are always estimated before the PI of the other derivatives [76].

Regression model

The identification of $\{X_{(\cdot)}', Y_{(\cdot)}', N_{(\cdot)}'\}$ is seen as the identification of linear regression models. The linear regression models in the surge, sway, and yaw directions are formulated in the discrete forms with the interval of the sampling time h as:

$$\Delta u(k+1) - \Delta u(k) = AX \tag{5.4}$$

$$\Delta v(k+1) - \Delta v(k) = BY \tag{5.5}$$

$$\Delta r(k+1) - \Delta r(k) = CN \tag{5.6}$$

(k) and $(k+1)$ denote velocities at the certain time step and those at the one-step ahead. $A = [a_1, ..., a_{16}]_{1\times 16}$, $B = [b_1, ..., b_{22}]_{1\times 22}$, and $C = [c_1, ..., c_{22}]_{1\times 22}$ are the parameter vectors to be identified. The input vectors of the linear regression models X, Y, and N are given as:

$$
\begin{aligned}
X = [&\Delta u(k)U(k), \Delta u^2(k), \Delta u^3(k)/U(k), \Delta v^2(k), \\
&\Delta r^2(k)L^2, \Delta\delta^2(k)U(k)^2, \Delta\delta^2(k)\Delta u(k)U(k), \\
&\Delta v(k)\Delta r(k)L, \Delta v(k)\Delta\delta(k)U(k), \Delta v(k)\Delta\delta(k)\Delta u(k), \\
&\Delta u(k)\Delta v(k)^2/U(k), \Delta r^2(k)L^2/U(k), \\
&\Delta u(k)\Delta v(k)\Delta r(k)L/U(k), \Delta r(k)\Delta\delta(k)LU(k), \\
&\Delta u(k)\Delta r(k)\Delta\delta(k)L, U^2(k)]_{1\times 16}^T
\end{aligned} \tag{5.7}
$$

$$
\begin{aligned}
Y = N = [&U^2(k), \Delta u(k)U(k), \Delta u^2(k), \Delta v(k)U(k), \Delta r(k)U(k)L, \\
&\Delta\delta(k)U^2(k), \Delta v^3(k)/U(k), \Delta\delta^3(k)U^2(k), \\
&\Delta v^2(k)\Delta r(k)L/U(k), \Delta v^2(k)\Delta\delta(k), \Delta v(k)\Delta\delta^2(k)U(k), \\
&\Delta\delta(k)\Delta u(k)U(k), \Delta v(k)\Delta u(k), \Delta r(k)\Delta u(k)L, \\
&\Delta\delta(k)\Delta u^2(k), \Delta r^3(k)L^3/U(k), \Delta v(k)\Delta r^2(k)L^2/U(k), \\
&\Delta v(k)\Delta u^2(k)/U(k), \Delta r(k)\Delta u^2(k)L/U(k), \\
&\Delta r(k)\Delta\delta^2(k)LU(k), \Delta r^2(k)\Delta\delta(k)L^2, \\
&\Delta r(k)\Delta v(k)\Delta\delta(k)L]^T_{1\times 22}
\end{aligned} \tag{5.8}
$$

In the PI, the left sides of (5.4) - (5.6) and input vectors are given. The PI algorithm finds the optimum A, B, and C so that they satisfy (5.4) - (5.6) well. Once they are identified, 16, 22, and 22 hydrodynamic derivatives for three directions $\{X'_{(\cdot)}, Y'_{(\cdot)}, N'_{(\cdot)}\}$ are further derived as:

$$
X'_{(\cdot)} = \frac{L(m' - X'_{\dot{u}})}{h} A \tag{5.9}
$$

$$
\begin{bmatrix} Y'_{(\cdot)} \\ N'_{(\cdot)} \end{bmatrix} = \begin{bmatrix} \dfrac{(I'_z - N'_{\dot{r}})h}{SL} & -\dfrac{(m'x'_G - Y'_{\dot{r}})h}{SL} \\ -\dfrac{(m'x'_G - N'_{\dot{v}})h}{SL^2} & \dfrac{(m' - Y'_{\dot{v}})h}{SL^2} \end{bmatrix}^{-1} \begin{bmatrix} B \\ C \end{bmatrix} \tag{5.10}
$$

where $S = (m' - Y'_{\dot{v}})(I'_z - N'_{\dot{r}}) - (m'x'_G - Y'_{\dot{r}})(m'x'_G - N'_{\dot{v}})$. m' and x'_G represent the non-dimensional mass and longitudinal position of the ship's center of gravity, respectively.

5.1.2 Model refinement

Here, the integral part of the proposed framework is introduced; the source ship (*where do we transfer knowledge from?*) and the algorithm (*how do we transfer knowledge?*)

Source ship; The proposed framework employs source parameters as a knowledge foundation and safeguard in data-driven calibration. Thereby, a ship is eligible for being a source ship based on an agreement of designers and stakeholders. Such an agreement is derived from their domain knowledge, experience, and how much risk they take in the project. Traditionally, the following factors support stakeholders to have sufficient confidence in the eligibility of the source ship.

- Two ships must belong to the same ship category.
- The similarity of two ships must be checked based on a ship characteristic vector as [18] did.
- Designers must check the performance of the source parameters when being applied to the collected dataset of the target ship.

Algorithm; The linear regression model has a general form:

$$
\hat{\boldsymbol{y}} = \hat{\beta}_0 + \sum_{j=1}^{p} \hat{\beta}_j \boldsymbol{x}_j \tag{5.11}
$$

where $\hat{\boldsymbol{y}}_{1\times m}$ denotes the output vector, $\boldsymbol{X} = [\boldsymbol{x}_1, ..., \boldsymbol{x}_p]_{p\times m}$ is the input matrix, $\hat{\beta}_0$ is the intercept, $\hat{\boldsymbol{\beta}} = [\hat{\beta}_1, ..., \hat{\beta}_p]_{1\times p}$ is the parameter vector of the regression model, m is the number of data samples, and p is the number of the input dimension. We assume the parameter vector of the source model $\boldsymbol{\beta}'^s$ (the source parameters) is available and that for the target ship $\hat{\boldsymbol{\beta}}'$ (the target parameters) needs to be identified. If some parameters are removed from (5.1)-(5.3) in the source model, such parameters are not included in the knowledge transfer. Usually, the input variable $\boldsymbol{x}_j$ is z-score normalized so that the penalization of $\hat{\boldsymbol{\beta}}$ is applied in the same scaling over the input variables as:

$$\boldsymbol{x}'_j = \frac{\boldsymbol{x}_j - \mu_j}{\sigma_j} \tag{5.12}$$

where μ_j and σ_j are the mean and standard deviation of the input variable in the training dataset. Then, we define $\boldsymbol{X}' = [\boldsymbol{x}'_1, ..., \boldsymbol{x}'_j, ..., \boldsymbol{x}'_p]_{p\times m} = [\boldsymbol{x}'_1, ..., \boldsymbol{x}'_i, ..., \boldsymbol{x}'_m]^T_{m\times p}$ as the z-score normalized input matrix. A linear regression model is written as:

$$\hat{\boldsymbol{y}} = \hat{\beta}'_0 + \sum_{j=1}^{p} \hat{\beta}'_j \boldsymbol{x}'_j \tag{5.13}$$

where we define:

$$\hat{\beta}'_j \equiv \hat{\beta}_j \sigma_j \tag{5.14}$$

$$\hat{\beta}'_0 \equiv \sum_{j=1}^{p} \hat{\beta}_j \mu_j + \hat{\beta}_0 \tag{5.15}$$

We estimate $\hat{\beta}'_0 = \sum_{i=1}^{m} y_i$ in the training dataset, where $\boldsymbol{y} = [y_1, ..., y_m]$ is the target vector in the training dataset. Under the assumption that the residuals between the target vector $\boldsymbol{y}$ and $\hat{\boldsymbol{y}}$ are normally distributed in addition to $\hat{\beta}_0 = 0$ in (5.4) - (5.6), this estimation is equivalent to the definition (5.15). The source parameters are converted as $\beta'^s_j \equiv \beta^s_j \sigma_j$ to fit this z-score normalization. Then, by using $\hat{\beta}'^s_j$ as a knowledge foundation, the refined parameter vector $\hat{\beta}'_j$ for the target ship is formulated as:

$$\begin{aligned}\hat{\boldsymbol{\beta}}' = \arg\min_{\boldsymbol{\beta}'} \frac{1}{m} \sum_{i=1}^{m} (\boldsymbol{\beta}'^T \boldsymbol{x}'_i - \Delta y_i)^2 \\ + \lambda(\alpha ||\boldsymbol{\beta}'||_2^2 + (1-\alpha)||\boldsymbol{\beta}' - \boldsymbol{\beta}'^s||_2^2)\end{aligned} \tag{5.16}$$

where $\lambda \geq 0$ is a hyperparameter for determining the regularization strength and $0 \leq \alpha \leq 1$ is a hyperparameter that balances the source-parameter vector $\boldsymbol{\beta}'^s$ and the zero vector in the regularization. $\Delta\boldsymbol{y} = \boldsymbol{y} - \hat{\beta}'_0$. $\alpha = 1$ and $\lambda = 0$ correspond to the plain ridge regression and linear regression, respectively. (5.16) is equivalent to:

$$\hat{\boldsymbol{\beta}}' = (\boldsymbol{X}'^T \boldsymbol{X}' + m\lambda \boldsymbol{I})^{-1} (\boldsymbol{X}'^T \boldsymbol{\Delta y} + m\lambda(1-\alpha)\boldsymbol{\beta}'^s) \tag{5.17}$$

thereby, $\hat{\boldsymbol{\beta}}'$ is derived analytically. In (5.16), α and λ are hyperparameters that characterize the l_2 regression terms. As mentioned, ranges of hyperparameter tuning are

Table 5.1: Specifications of two ships.

	Source ship	**Target ship**
Ship name	Mariner ([81])	SR108 ([82])
Length between perpendiculars	160.9 m	175.0 m
Breadth	23.2 m	25.4 m
Design draft	8.2 m	8.5 m
Design displacement	18541 m^3	21222 m^3
Propeller diameter	6.706 m	6.533 m
The number of propellers	1	1
The number of blades	4	5

defined by experts to balance their confidence in knowledge against data-driven calibration. In (5.16), α and λ are hyperparameters that characterize the l_2 regression terms. As mentioned, ranges of hyperparameter tuning are defined by experts to balance their confidence in knowledge against data-driven calibration. In such ranges, they are tuned so that it minimizes the Root Mean Squared Error (RMSE) in the validation dataset, which stays away from the training dataset.

5.1.3 Validation experiment

In the case study, a simulation experiment transferring knowledge between two ships was conducted to present an example case where the source model was helpful to find a robust-and-accurate model for the target ship. The case study performed the model refinement of the Abkowitz model of a 161m-length container Mariner class vessel [81], which was referred to as a source ship, for better fitting maneuverings of a 175m-length container ship SR108 [82], which was referred to as a target ship. For the source ship, an Abkowitz model has been widely-used and validated. It is a source model and its parameters are source parameters in the case study. For the target ship, we assumed only a limited dataset was available. The source and target ships were highly similar, however, we found that the source parameters had room to be calibrated for maneuvers of the target ship. In the case study, we aimed to reduce 50% of the single-step-ahead velocity prediction error in the validation dataset in three directions. In practice, such criteria need to be carefully agreed upon with stakeholders.

Experiment setting

Source and target ships; A comparison of basic specifications of source and target ships are shown in Tab. 5.1. The source model for the source ship was for the operation speed of 15.0 knots. A characteristic vector l is defined based on domain understanding of ship design:

$$l = [C_b, L_{pp}/B, B/T, L_{pp}/\Delta^{1/3}, Ar/(L_{pp}T), D_p/T] \tag{5.18}$$

where C_b is the block coefficient, L_{pp} is the length between perpendiculars, B is the breadth, T is the draught, Δ is the displacement, Ar is the rudder area, and D_p is the diameter of the propeller. $l^s = [0.59, 6.94, 2.82, 6.08, 0.023, 0.81]$ was for the source ship and $l^t = [0.56, 6.89, 2.99, 6.32, 0.022, 0.77]$ was for the target ship. Hence, two ships, which were categorized in the same ship type, have very similar characteristic vectors. In fact, using the source parameters for maneuvers of the target ship reproduced the trend

of the target ship's response well, however, a data-driven calibration was still necessary for better performance.

Dataset; On the MSS simulator [83], multiple maneuvers of the target ship were conducted for refining the source parameters in the case study. A zigzag maneuver, 12.5° and -12.5° turning circle maneuvers were generated. Each maneuver had a 650s time series of positions, heading, velocities, and rudder angles saved in 2Hz with the artificial measurement noise. To check if the proposed framework finds a robust-and-accurate model for different types of maneuvers of the target ship, three different case studies were conducted by taking three operation speeds $U_0 = 16.3$, 20.4, and 24.7 knots for low, middle, and high-speed maneuvering case studies, respectively. To evaluate the sensitivity of performances against having different training datasets, five training datasets were prepared for each case study by preparing five different zigzag maneuvers. for the case study with low-speed maneuverings. In total, three case studies (high, middle, low-speed maneuvers of the target ship) were conducted, and each case study had five different datasets for training and validation. The final evaluation was performed in a test dataset, which included 7.5° and -7.5° turning circle maneuvers. Maneuvers with 7.5° and -7.5° rudder angles were conducted only in the test dataset. Maneuvers in the test dataset were conducted at the corresponding high, middle, and low speeds for each case study. During the hyperparameter tuning, 70% of samples were employed for the training and the remaining samples were kept untouched for validation.

Results

To evaluate the benefit of the proposed method, four baseline algorithms were implemented in addition to prediction made by using the source parameters without data-driven calibration.

- Model refinement (MR): Results made by the present study.
- Source parameters (SP): Results with the source parameters of the source ship without any data-driven calibration.
- Ridge regression (RR): Results made by the ridge regression algorithm.
- Lasso regression (LR): Results made by the lasso regression algorithm.
- Elastic net (EN): Results made by the elastic-net algorithm.
- Support vector regression (SVR): Results made by the support vector regression algorithm.

Hyperparameter tuning for baselines was done by hyperparameter tuning framework optuna [48], of which details are found in Appendix.E.

Table 5.2: A comparison of performances of the present model refinement (MR) and five baseline algorithms for two turning circle maneuvers in the test dataset. Steady Turning Radius (STR), Maximum Transfer (MT), Maximum Advance (MA), and Velocity Loss on Steady turn (VLS) defined by the ITTC are compared. The average absolute error is that made by five different training datasets. The standard deviation of absolute error is its standard deviation. N/A indicates one or more trajectories made by five training datasets result in physically unreasonable trajectories. The red numbers show the best performance against the other frameworks for each metric of each maneuver.

			True	Average absolute error μ(%)						Standard deviation of absolute error σ(%)				
				SP	**MR**	LR	RR	EN	SVR	**MR**	LR	RR	EN	SVR
Case 1 Low speed	7.5 ° turning	STR (m)	824.9	10.3	3.6	12.7	N/A	11.8	N/A	0.4	6.3	N/A	6.5	N/A
		MT (m)	1683.2	6.1	4.5	5.4		5.6		0.4	5.4		5.0	
		MA (m)	1099.2	6.5	9.6	6.2		6.3		0.3	3.2		3.1	
		VLS (m/s)	1.3	38.2	0.9	51.1		56.2		0.6	19.1		10.0	
	-7.5 ° turning	STR (m)	817.1	28.0	5.4	16.5	10.7	16.5	17.6	0.2	5.1	7.5	5.9	7.1
		MT (m)	1683.2	22.9	2.9	10.3	21.9	10.9	12.3	0.2	8.6	17.2	8.5	9.3
		MA (m)	1099.2	9.3	2.6	9.4	16.9	9.6	14.0	0.2	5.7	10.0	5.7	9.6
		VLS (m/s)	1.3	49.5	9.0	56.7	45.9	60.9	64.8	0.8	9.9	4.4	5.2	7.3
Case 2 Middle speed	7.5 ° turning	STR (m)	760.7	19.6	1.0	25.8	32.1	24.8	N/A	0.3	15.2	14.2	13.1	N/A
		MT (m)	1558.5	14.6	1.8	19.2	30.7	18.8		0.3	6.4	14.0	5.7	
		MA (m)	1039.8	1.9	8.3	10.3	16.4	10.2		0.2	7.8	7.4	7.4	
		VLS (m/s)	1.8	43.8	2.3	43.2	26.0	29.6		0.6	35.0	11.1	12.8	
	-7.5 ° turning	STR (m)	755.3	38.5	7.0	14.5	N/A	13.6	27.1	0.4	4.7	N/A	4.1	18.1
		MT (m)	1558.5	32.8	4.7	15.1		14.9	25.3	0.5	4.2		4.2	29.6
		MA (m)	1039.8	14.4	2.5	9.2		9.0	22.4	0.5	2.6		2.5	15.5
		VLS (m/s)	1.8	54.1	10.4	51.3		34.1	55.3	0.9	31.7		15.0	12.2
Case 3 High speed	7.5 ° turning	STR (m)	680.0	33.8	1.1	14.0	77.9	14.0	N/A	0.3	6.3	60.6	5.1	N/A
		MT (m)	1375.3	29.8	2.4	11.9	39.1	11.8		0.3	6.9	29.8	7.2	
		MA (m)	962.3	5.1	7.3	6.5	8.1	6.3		0.2	2.5	5.2	2.1	
		VLS (m/s)	2.5	50.9	3.3	65.3	24.9	56.5		0.8	22.2	17.2	12.1	
	-7.5 ° turning	STR (m)	676.3	54.6	9.0	12.4	41.9	12.9	N/A	0.4	6.0	25.9	5.5	N/A
		MT (m)	1375.3	50.4	8.8	11.3	34.0	11.8		0.4	7.3	20.8	6.8	
		MA (m)	962.3	22.6	1.8	5.3	11.6	5.5		0.3	2.5	6.1	2.6	
		VLS (m/s)	2.5	60.0	10.6	64.5	27.8	56.0		0.9	22.8	13.9	12.1	

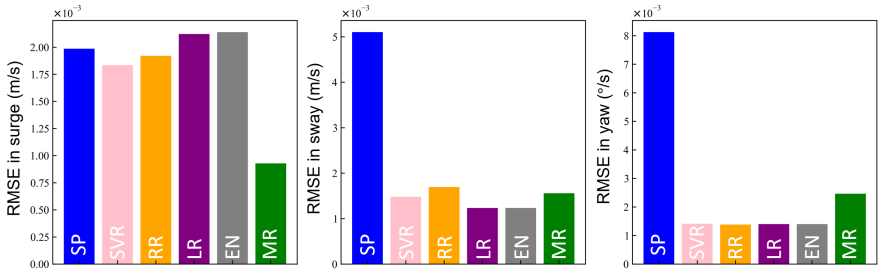


Figure 5.1: RMSEs in the single-step-ahead velocity prediction in the surge, sway, and yaw directions made by the present MR, SP, and four baseline algorithms. Results here are for the low-speed case study with one of the five training datasets. Note that five lines with five different training datasets for MR highly overlap.

For three directions, firstly, tuning ranges of hyperparameters were set to $\alpha \in [0.0, 0.1]$ and $\lambda \in [10, 10^{10}]$ that highly constrain parameters towards the main parameters. Results with the tuned hyperparameters are shown hereinafter. Fig. 5.1 show RMSEs of single-step-ahead velocity prediction in the surge, sway, and yaw directions in the validation dataset. In Fig. 5.1, results using one of the training datasets of low-speed maneuverings are shown. In the surge direction, having a knowledge foundation in MR was found to be powerful in reducing validation loss. As shown in Fig. 5.1, whereas MR performed the best in the validation dataset with highly constrained parameter change from SP, data-based frameworks (RR, SVR, LR, and EN) resulted in much poorer performances with notably deviated parameters from SP. This result presented a challenge in solely relying on data in the case study since the dataset had limited excitation in the surge direction. In the sway direction, MR achieved an equivalent performance with the much less number of parameters, of which change was highly constrained towards SP, than that identified by the data-based frameworks. It can be seen as a big advantage since a simpler model is more robust than more complex models with equivalent performance. In the yaw direction, Fig. 5.1 shows a different trend from results in the other directions. MR calibrated SP well, however, models with very different parameters from the source parameters, which were built in data-based frameworks, performed better in the validation dataset than MR. Thereby, by broadening ranges of hyperparameter tuning and less prioritizing a knowledge foundation, the validation loss of MR could be reduced. Here is where experts introduce their confidence in knowledge against data-driven calibration. Given that a dataset was rather limited in this case study and MR achieved an improvement criterion of 50%, this case study did not allow models to further be calibrated by data by broadening ranges of hyperparameter tuning. As sufficient improvements were achieved by adjusting the source parameters, the same ranges of hyperparameter tuning were applied to all case studies. Hence, the proposed framework offered such a conservative option within the confidence range around the source parameters rather than forgetting it and building up a new model solely relying on data.

A summary of results is listed in Tab. 5.2. In the test maneuvers, this table compares prediction errors in four metrics defined by the ITTC for the turning-circle maneuvers:

Steady Turning Radius (STR), Maximum Transfer (MT), Maximum Advance (MA), and Velocity Loss On Steady turn (VLS). For each metric, the error rate e (%) was calculated for all predictions.

$$e = |\frac{m_{\text{true}} - m_{\text{pred}}}{m_{\text{true}}}| \times 100 \tag{5.19}$$

where m_{true} and m_{pred} are true and predicted values for each metric. For each case study, five different training datasets were used. In Tab. 5.2, values in the columns for the average absolute error show μ for each metric:

$$\mu = \frac{1}{K}\Sigma_{i=1}^{K} e_i \tag{5.20}$$

where $K = 5$ is the number of the training datasets and e_i is e for prediction made by the i-th training dataset. Please note that $\mu = e$ for SP since it does not use any dataset. Values in the columns for the standard deviation of absolute error show σ for each metric:

$$\sigma = \sqrt{\frac{1}{K}\Sigma_{i=1}^{K}(e_i - \mu)^2} \tag{5.21}$$

Note that $\sigma = 0$ for SP since it does not use any dataset, thereby, it is not shown in the table. Small μ and σ show that corresponding algorithms achieve the average good performance over experiments using five different training datasets and such a performance is stably accomplished. "N/A" indicates that one or more models in the five models identified by using five training datasets ended up with physically unreasonable trajectories. If the heading does not either monotonically decrease or increase, the trajectory was considered physically unreasonable. First, we focus on the average absolute errors. In terms of almost all metrics of almost all test maneuvers, SP made much prediction error. In particular, when the target maneuver was high-speed, their prediction errors were found to be large since the source parameters were not well-adjusted to it. On the other hand, such large prediction errors were notably mitigated in the present MR. As RR and SVR did not perform a feature selection, the identified models were not robust. LR and EN selected only informative features with the l_1 loss function, thereby, the identified models were robust and stable without having redundant features in the regression models. However, the present MR outperformed SP and four baseline algorithms in almost all metrics of almost all test maneuvers. In light of the standard deviation of the absolute errors, the robustness of the performance of the present MR against having different training datasets were clearly shown. It is reasonable since MR appreciates the source model and parsimoniously exploits the target datasets. Four baseline algorithms were found to be very sensitive to the selection of training datasets as they were fully dependent on data without having the confidence range around the source parameters.

5.1.4 Discussion

In the case study, the PI of the Abkowitz model for the target ship was conducted. Due to the high similarity between the ship characteristic vectors of the two ships, their resulting models were expected to be highly relevant. However, data-driven refinement of the source parameters was necessary to satisfy the performance criteria for maneuvers of the target ship. In the case study, the proposed framework was found to be powerful in

easily finding a refined model with good robustness and generalization performance, by prioritizing a safe zone of calibration derived by experts' confidence in parameter transfer from the source ship. We prepared five different training datasets in the case study. The proposed framework found a robust-and-accurate model and stably accomplished a great performance against having five different training datasets. On the contrary, without having the parameters of the source ship, model performance was found to highly fluctuate.

In the case study, the proposed framework showed better performance in the test dataset than traditional data-based frameworks. This result presented an example case where the proposed framework was powerful in easily finding a robust-and-accurate model within a safe zone drawn based on the source parameters without being sensitive to the experiment design. Nevertheless, it does not show an absolute limitation of data-based frameworks. Their performance could be improved by devoting more effort for the data design, model design, PI algorithm design. However, this study focuses on cases where we have less room to do so and need to efficiently deliver the target model. It would perform a complementary role to data-based frameworks in such cases rather than always replacing them. Especially for delivering models integrated into safety-critical systems, data-based frameworks remain important since collecting much data is the only way to get closer to the global optimum parameters.

Typically, shipyards build many ships with similar specifications, so it is expected that similar ship models may be found. Moreover, when constructing models for different loading and water-depth conditions, it is possible to consider a standard-condition model as a similar-ship model. Therefore, the scope of application for this framework is considered to be wide. On the other hand, it does not fully replace the traditional framework. When there is no reliable similar-ship model, the proposed framework cannot be used. To construct accurate and reliable models, either a knowledge foundation or data collected through careful experiments is required.

In this field, a validation study can not be conducted comprehensively. The ITTC 29th Manoeuvring Committee ([84]) asked different institutes how they validate simulation models. A majority of answers were validations based on previous experience with other ships or benchmark data, expert judgment, and comparing the trajectories with free-running tests. Any of them does not have unified criteria. In practice, experts make an agreement so that they satisfy their confidence in the model's validity. Thereby, it is reasonable that the proposed framework introduces their decision into selecting a final model from rival models based on their understanding of the relevance of similar ships' dynamics. This study helps such a practice be efficiently conducted. As a matter of fact, it contributed to building a robust-and-accurate model for the target ship. Such a decision is not randomly made but based on experts' experience and how much risk the project takes.

In this study, we addressed the risk of conducting a global parameter search without any knowledge anchor. By using a reliable similar ship model to locally update the model, the knowledge connection between ships can be established and the risk can be reduced. To demonstrate this concept, this study dealt with a 3DoF maneuvering model in calm water. However, nowadays, models that include external-force models such as waves are considered more practical. The applicability of the proposed framework to such advanced models has not been demonstrated in this paper. However, given that the

external force term is a parametric model that relates to a ship's mechanical performance, this framework is also expected to contribute to reducing the number of experiments and risks associated with creating an external-force model. On the other hand, defining similarities between ships becomes more complicated in models that consider external forces. For example, even if the maneuvering characteristic is similar, there may be a significant deviation in the dynamic positioning characteristic. Regarding the specific definition of similarity, further discussion will be necessary in our future study.

5.2 Chapter summary

In this chapter, the author proposed a method for finding a robust-and-accurate model for a target ship by using a model of a similar ship that experts and stakeholders believe in their similarities. In this field, data collection is challenging and time-consuming. In particular, when quickly obtaining a model, data is often insufficient and exposed to noises, as is the case in this dissertation. In such cases, as the author presented, it seems a wise framework to take the foundation model from a similar ship and select an updated model with acceptable accuracy from those that are close to the foundation model. This framework is proven to be helpful to make the model development easier and stabler for achieving high accuracy. This chapter shows the promising direction of the DE framework towards building a network of various ship models stemming from one foundation model, which could be seen as "a family of ship models".

6 Conclusion and further work

This dissertation explored agile methods for delivering dynamic models for ship motion prediction. The author proposed a new framework: the DE framework which enhances the physics-based model in a data-driven manner for better performance. The framework involves the foundation model, where experts locate their understanding and reliability, and the data-driven branch, which utilizes advanced machine learning models to inductively fit datasets. They stem from distinguished modeling principles, with the former being physics-driven and the latter being data-driven. Ships are complex multi-scale and multi-disciplinary systems, and it is not possible to quickly deliver accurate dynamic models by relying upon either of these principles. The present framework attempts to maximize performance by respecting the roles of both modeling principles rather than attempting to replace one with the other. In this dissertation, we proposed new techniques to do so and demonstrated the promise of such a framework respecting knowledge connection when delivering ship dynamic models.

6.1 Summary of contributions

The goal of this dissertation is to explore the enabler of the DE framework, as outlined in RO1, and to offer accurate ship dynamic models in a more agile manner. In Section 2, possible methods for achieving this goal were over-viewed through relating to multi-disciplinary previous works. Additionally, the papers in this dissertation contribute to satisfying RO1 by proposing and investigating different techniques to realize the DE framework from various perspectives. Chapter 3 presents four case studies to accomplish RO2. They demonstrate that the DE framework can be realized in various forms depending on pragmatic constraints and its application contexts. In all cases, the successful enhancement of the foundation model resulted in readily offering a more accurate model. Chapter 4 investigates how the foundation model and data benefit enhanced performance in an experimental way. Knowledge transfer between similar ships is investigated in Chapter 5 with simulation case studies.

The main contributions of this dissertation are as follows:

- Present a framework to use data-driven models for enhancing the physics-based model serving as a knowledge foundation. Clarify its overview and possible enablers. It is related to RO1.
- Implement techniques of the DE framework for rapidly reflecting the full-scale data to better performance while utilizing a knowledge foundation from the foundation model. It is related to RO2.
- Propose a novel method for bridging similar ships' dynamics for identifying model

parameters more easily and stably while establishing a knowledge connection. It is related to RO3.

- Investigate how the foundation model and data benefit enhanced performance in the DE framework. Extensive experimental studies are presented, related to RO4.

6.2 Summary of publications

The summary of publications is as follows:

Paper I presents a method for the DE framework that uses the foundation model to predict a trajectory and correct it in a data-driven manner with a neural network. The assumption of constant commands over the prediction horizon has been previously used. However, it is not always valid for real-life data. In paper I, the proposed model efficiently handles future command changes without sacrificing the training efficiency much. In the simulation environment, it was proven to work under different environmental disturbances, demonstrating its performance and robustness. This allows for efficient learning in a way that highly isolates the foundation model and data with consideration for future command assumptions over the prediction horizon.

Paper II presents a novel "pre-training" method for the DE framework using the foundation model and dataset for predicting ship maneuvering motion. In the DE framework, as in other papers included in this dissertation, the straightforward method is to directly embed or modify the foundation model by using a newly-available dataset. However, there exist situations where we may not be able to do so due to rights or technical challenges. In such cases, as proposed in Paper II, the foundation model on a simulator can be used for data augmentation. After sufficiently transferring knowledge from the foundation model, the full-scale dataset can further benefit the model for better performance. In the validation study, an augmented dataset was created by using the foundation model on the co-simulation platform. By warming up the neural-network ship motion predictor with the dataset, better accuracy was achieved by further using a limited full-scale dataset.

Paper III presents a new method of the DE framework for ship maneuvering prediction during docking operations. Docking operations are known as one of the most complex and nonlinear operations in which ships make a sharp deceleration while rapidly turning. It has been challenging to build a highly-accurate foundation model in a parametric manner for such dynamics. In Paper III, a method was proposed to calibrate the velocity-prediction performance of the foundation model with neural networks. This allows for building a non-parametric data-driven branch for docking operations on top of the foundation model. Validation experiments were conducted using a full-scale dataset of docking operations, confirming the high performance of the proposed method.

Paper IV investigates the benefit of the foundation model and dataset, which are two key pillars in the DE framework, on enhanced performance. The problem awareness of this dissertation is the fact that both of these pillars are not often satisfactory in terms of quality and quantity. Therefore, it is crucial to understand the impact of these factors on enhanced performance. This may allow us to compromise on the performance of the foundation model or data quantity for providing models more quickly. Paper IV employed a DE framework, which is an extended work from Paper I, to provide one solution to this question through comprehensive simulation and full-scale experiments.

The results showed that the performance of the foundation model and the quantity of the dataset support complementary enhanced performance.

Paper V proposes a method for the DE framework to bridge the dynamics of similar ships. In the proposed method, the parameter set of a physics-based model for a new ship is obtained from the vicinity of that for a similar ship. It aims at gaining the benefit from a dataset of the new ship while anchoring the knowledge foundation transferred from the similar ship. The validation study was conducted by employing two cargo ships on the simulation platform. The proposed Bayesian method showed that by carefully selecting similar ships based domain knowledge, it found a robust-and-accurate model easily and stably. With the proposed method, it may be possible to build a family of ship models that collaborate on the same knowledge foundation by sharing a single foundation model among multiple ship models.

Paper VI presents a method to apply the DE framework to predict ship motion during DP operations using an onsite wave radar. To the best of the author's knowledge, real-world experiments for such a task are very limited in previous works and have relied on physics-based models. However, it is an arduous process to develop such an accurate foundation model. Therefore, in paper VI, the author corrected the imbalance of the full-scale operation datasets and enhanced the foundation model by mapping it using a linear regression model. This demonstrates that the DE framework is useful not only for maneuvering but also for wave-frequency motions.

6.3 Future work

This dissertation conceptualized the data-driven enhancement to the ship dynamic model for motion prediction. The author focused on proposing new techniques for priority problem settings and investigating important research questions in the DE framework. It demonstrated the promising direction of the DE framework, however, it is not a complete elucidation of it. The below bullet points provide suggestions for how the presented research may be extended.

- The pillars of the proposed DE framework are the foundation model and its data-driven branches. As introduced in paper IV, they can work complementarily to quickly develop a satisfactory enhanced model by compromising both of them to some extent. In future work, it would be key research topics (1) how to rapidly obtain the foundation model with decent accuracy and (2) how to collect high-quality data at a low cost. Such topics may become as crucial as how we realize the DE framework itself.

- This dissertation did not address a topic of continuous learning, where new data is continuously available and the model is updated online. However, ship dynamics may perpetually change due to external disturbances or loading conditions. In such cases, separating the foundation model and its data-driven branches may enable more effective continuous learning.

- When there are many foundation models available, it may be better to select appropriate models or ensembling from them, rather than enhancing them through data-driven methods. Such a data-driven model selection or ensemble remains to be investigated.

- In this dissertation, the contribution of data to enhanced performance has been mainly discussed from the viewpoint of 'data volume'. On the other hand, real-life data often present issues related to 'data quality', such as noises and anomalies in datasets. This dissertation indirectly and practically addressed such issues by using real-world datasets. To reach more practical conclusions for the proposed framework, in the future, more focused research on this 'data quality' would be necessary.

References

[1] G. Zhang and V. V. Thai, "Expert elicitation and Bayesian Network modeling for shipping accidents : A literature review," *Safety Science*, vol. 87, pp. 53–62, 2016.

[2] Y. Huang, L. Chen, P. Chen, R. R. Negenborn, and P. H. A. J. M. V. Gelder, "Ship collision avoidance methods : State-of-the-art," *Safety Science*, vol. 121, no. April 2019, pp. 451–473, 2020.

[3] Norges teknisk-naturvitenskapelige universitet, "Research vessel: R/V Gunnerus."

[4] L. Huang, B. Pena, Y. Liu, and E. Anderlini, "Machine learning in sustainable ship design and operation: A review," *Ocean Engineering*, vol. 266, 12 2022.

[5] "Standards for Ship Maneuverability," tech. rep., International Maritime Organization, 12 2002.

[6] R. Skulstad, G. Li, T. I. Fossen, B. Vik, and H. Zhang, "Dead Reckoning of Dynamically Positioned Ships: Using an Efficient Recurrent Neural Network," *IEEE Robotics and Automation Magazine*, vol. 26, no. 3, pp. 39–51, 2019.

[7] R. Li, J. Huang, X. Pan, Q. Hu, and Z. Huang, "Path following of underactuated surface ships based on model predictive control with neural network," *International Journal of Advanced Robotic Systems*, vol. 17, no. 4, pp. 1–10, 2020.

[8] S. Li, J. Liu, R. R. Negenborn, and Q. Wu, "Automatic Docking for Underactuated Ships Based on Multi-Objective Nonlinear Model Predictive Control," *IEEE Access*, vol. 8, pp. 70044–70057, 2020.

[9] Y. Liu, W. Duan, L. Huang, S. Duan, and X. Ma, "The input vector space optimization for LSTM deep learning model in real-time prediction of ship motions," *Ocean Engineering*, vol. 213, no. August, p. 107681, 2020.

[10] T. I. Fossen, *Handbook of marine craft hydrodynamics and motion control*. Wiley, 2011.

[11] Nomoto.K, Taguchi.T, Honda.K, and Hirano.S, "On the steering qualities of ships," *International Shipbuilding Progress*, vol. 4, pp. 07–01, 7 1957.

[12] A. M.A., "Lectures on Ship Hydrodynamics - Steering and Maneuverability," tech. rep., Hydro- and Aerodynamic's Laboratory, Lyngby, 1964.

[13] T. I. Fossen, "A nonlinear unified state-space model for ship maneuvering and control in a seaway," *International Journal of Bifurcation and Chaos in Applied Sciences and Engineering*, vol. 15, no. 9, pp. 2717–2746, 2005.

[14] H. Yasukawa and Y. Yoshimura, "Introduction of MMG standard method for ship maneuvering predictions," *Journal of Marine Science and Technology (Japan)*, vol. 20, no. 1, pp. 37–52, 2015.

[15] Maneuvering committee, "Final report and recommendations to the 23rd ITTC," tech. rep., The international towing tank conference, Venice, 2002.

[16] The specialist committee on Esso Osaka, "Final report and recommendations to the 23rd ITTC," tech. rep., The international towing tank conference, 2002.

[17] G. De Masi, F. Gaggiotti, R. Bruschi, and M. Venturi, "Ship motion prediction by radial basis neural networks," *IEEE SSCI 2011 - Symposium Series on Computational Intelligence - HIMA 2011: 2011 IEEE Workshop on Hybrid Intelligent Models and Applications*, pp. 28–32, 2011.

[18] B. Mei, L. Sun, and G. Shi, "White-Black-Box Hybrid Model Identification Based on RM-RF for Ship Maneuvering," *IEEE Access*, vol. 7, pp. 57691–57705, 2019.

[19] H. Rong, A. P. Teixeira, and C. Guedes Soares, "Ship trajectory uncertainty prediction based on a Gaussian Process model," *Ocean Engineering*, vol. 182, no. December 2018, pp. 499–511, 2019.

[20] W. Y. Duan, L. M. Huang, Y. Han, and R. Wang, "IRF - AR model for short-term prediction of ship motion," in *Proceedings of the International Offshore and Polar Engineering Conference*, vol. 2015-Janua, (Kona, Big Island, Hawaii, USA), pp. 59–66, International Society of Offshore and Polar Engineers, 2015.

[21] B. Kawan, H. Wang, G. Li, and K. Chhantyal, "Data-driven Modeling of Ship Motion Prediction Based on Support Vector Regression," in *Proceedings of the 58th Conference on Simulation and Modelling (SIMS 58) Reykjavik, Iceland, September 25th – 27th, 2017*, vol. 138, pp. 350–354, 2017.

[22] C. Rudin, "Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead," *Nature Machine Intelligence*, vol. 1, no. 5, pp. 206–215, 2019.

[23] Y. Ichimura, D. Dalaklis, M. Kitada, and A. Christodoulou, "Shipping in the era of digitalization: Mapping the future strategic plans of major maritime commercial actors," *Digital Business*, vol. 2, p. 100022, 3 2022.

[24] L. Vonrueden, S. Mayer, K. Beckh, B. Georgiev, S. Giesselbach, R. Heese, B. Kirsch, M. Walczak, J. Pfrommer, A. Pick, R. Ramamurthy, J. Garcke, C. Bauckhage, and J. Schuecker, "Informed Machine Learning - A Taxonomy and Survey of Integrating Prior Knowledge into Learning Systems," *IEEE Transactions on Knowledge and Data Engineering*, pp. 1–19, 2021.

[25] A. Karpatne, G. Atluri, J. Faghmous, M. Steinbach, A. Banerjee, A. Ganguly, S. Shekhar, N. Samatova, and V. Kumar, "Theory-guided Data Science: A New Paradigm for Scientific Discovery from Data," *IEEE Transactions on Knowledge and Data Engineering*, vol. 29, pp. 2318–2331, 12 2017.

[26] G. E. Karniadakis, I. G. Kevrekidis, L. Lu, P. Perdikaris, S. Wang, and L. Yang, "Physics-informed machine learning," *Nature Reviews Physics*, vol. 3, no. 6, pp. 422–440, 2021.

[27] S. Cuomo, V. S. di Cola, F. Giampaolo, G. Rozza, M. Raissi, and F. Piccialli, "Scientific Machine Learning through Physics-Informed Neural Networks: Where we are and What's next," *Journal of Scientific Computing*, vol. 92, 1 2022.

[28] W. Ji, W. Qiu, Z. Shi, S. Pan, and S. Deng, "Stiff-PINN: Physics-Informed Neural Network for Stiff Chemical Kinetics," *Journal of Physical Chemistry A*, vol. 125, pp. 8098–8106, 9 2021.

[29] J. H. Lagergren, J. T. Nardini, R. E. Baker, M. J. Simpson, and K. B. Flores, "Biologically-informed neural networks guide mechanistic modeling from sparse experimental data," *PLoS Computational Biology*, vol. 16, 12 2020.

[30] L. H. Queiroz, F. P. Santos, J. P. Oliveira, and M. B. Souza, "Physics-Informed deep learning to predict flow fields in cyclone separators," *Digital Chemical Engineering*, vol. 1, 12 2021.

[31] S. Panigrahi, A. Nanda, and T. Swarnkar, "A Survey on Transfer Learning," *Smart Innovation, Systems and Technologies*, vol. 194, pp. 781–789, 2021.

[32] F. Zhuang, Z. Qi, K. Duan, D. Xi, Y. Zhu, H. Zhu, H. Xiong, and Q. He, "A Comprehensive Survey on Transfer Learning," *Proceedings of the IEEE*, vol. 109, no. 1, pp. 43–76, 2021.

[33] Y. Sajedi and M. Bozorg, "Robust estimation of hydrodynamic coefficients of an AUV using Kalman and H∞ filters," *Ocean Engineering*, vol. 182, no. January, pp. 386–394, 2019.

[34] M. T. Sabet, H. M. Daniali, A. Fathi, and E. Alizadeh, "Identification of an Autonomous Underwater Vehicle Hydrodynamic Model Using the Extended, Cubature, and Transformed Unscented Kalman Filter," *IEEE Journal of Oceanic Engineering*, vol. 43, no. 2, pp. 457–467, 2018.

[35] P. Cardenas and E. A. De Barros, "Estimation of AUV Hydrodynamic Coefficients Using Analytical and System Identification Approaches," *IEEE Journal of Oceanic Engineering*, vol. 45, no. 4, pp. 1157–1176, 2020.

[36] M. Alexandersson, W. Mao, and J. W. Ringsberg, "System identification of Vessel Manoeuvring Models," *Ocean Engineering*, vol. 266, p. 112940, 12 2022.

[37] M. Alexandersson, D. Zhang, W. Mao, and J. W. Ringsberg, "A comparison of ship manoeuvrability models to approximate ship navigation trajectories," *Ships and Offshore Structures*, 2022.

[38] M. von Stosch, R. Oliveira, J. Peres, and S. Feyo de Azevedo, "Hybrid semi-parametric modeling in process systems engineering: Past, present and future," *Computers and Chemical Engineering*, vol. 60, pp. 86–101, 2014.

[39] T. Wang, G. Li, L. I. Hatledal, R. Skulstad, V. Aesoy, and H. Zhang, "Incorporating Approximate Dynamics Into Data-Driven Calibrator: A Representative Model for Ship Maneuvering Prediction," *IEEE Transactions on Industrial Informatics*, pp. 1–1, 2021.

[40] P. W. van de Ven, T. A. Johansen, A. J. Sørensen, C. Flanagan, and D. Toal, "Neural network augmented identification of underwater vehicle models," *Control Engineering Practice*, vol. 15, no. 6, pp. 715–725, 2007.

[41] X. Cheng, G. Li, R. Skulstad, P. Major, S. Chen, H. Petter, and H. Zhang, "Data-driven uncertainty and sensitivity analysis for ship motion modeling in o ff shore operations," *Ocean Engineering*, vol. 179, no. January, pp. 261–272, 2019.

[42] L. Chen, Y. Hontoir, D. Huang, J. Zhang, and A. J. Morris, "Combining first principles with black-box techniques for reaction systems," *Control Engineering Practice*, vol. 12, pp. 819–826, 7 2004.

[43] P. Ïnik and I. Grabec, "Empirical modeling of antibiotic fermentation process using neural networks and genetic algorithms," tech. rep.

[44] R. Skulstad, G. Li, T. I. Fossen, B. Vik, and H. Zhang, "A Hybrid Approach to Motion Prediction for Ship Docking - Integration of a Neural Network Model into the Ship Dynamic Model," *IEEE Transactions on Instrumentation and Measurement*, vol. 70, 2021.

[45] S. Ben Taieb, A. Sorjamaa, and G. Bontempi, "Multiple-output modeling for multi-step-ahead time series forecasting," *Neurocomputing*, vol. 73, no. 10-12, pp. 1950–1957, 2010.

[46] P. Han, G. Li, X. Cheng, S. Skjong, and H. Zhang, "An Uncertainty-Aware Hybrid Approach for Sea State Estimation Using Ship Motion Responses," *IEEE Transactions on Industrial Informatics*, vol. 18, pp. 891–900, 2 2022.

[47] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Computation*, vol. 9, pp. 1735–1780, 11 1997.

[48] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, "Optuna: A next-generation hyperparameter optimization framework," in *KDD '19: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, (Anchorage, AK, USA), pp. 2623–2631, Association for Computing Machinery, 2019.

[49] J. Bergstra, D. Yamis, and D. Cox, "Making a science of model search: hyperparameter optimization in hundreds of dimensions for vision architectures," in *Proceedings of the 30th International Conference on Machine Learning*, vol. 28, (Atlanta, Georgia, USA), 2013.

[50] J. Bergstra, R. Bardenet, Y. Bengio, and K. Balazs, "Algorithms for Hyper-Parameter Optimization," in *Advances in Neural Information Processing Systems (NIPS 2011)*, pp. 1–9, 2011.

[51] C. Gomes, C. Thule, D. Broman, P. G. Larsen, and H. Vangheluwe, "Co-simulation: A survey," *ACM Computing Surveys*, vol. 51, 4 2018.

[52] F. Perabo, D. Park, M. K. Zadeh, O. Smogeli, and L. Jamt, "Digital Twin Modelling of Ship Power and Propulsion Systems: Application of the Open Simulation Platform (OSP)," *IEEE International Symposium on Industrial Electronics*, vol. 2020-June, pp. 1265–1270, 2020.

[53] L. I. Hatledal, Y. Chu, A. Styve, and H. Zhang, "Vico: An entity-component-system based co-simulation framework," *Simulation Modelling Practice and Theory*, vol. 108, no. September 2020, p. 102243, 2021.

[54] L. I. Hatledal, H. Zhang, and F. Collonval, "Enabling Python Driven Co-Simulation Models With PythonFMU," in *34th International ECMS Conference on Modelling and Simulation*, 2020.

[55] V. Hassani, D. Fathi, A. Ross, F. Sprenger, O. Selvik, and T. E. Berg, "Time domain simulation model for research vessel Gunnerus," in *Proceedings of the International Conference on Offshore Mechanics and Arctic Engineering - OMAE*, vol. 7, (St.John's, Newfoundland, Canada), 2015.

[56] K. Hasselmann, T. P. Barnett, E. Bouws, H. Carlson, D. E. Cartwright, K. Eake, J. A. Euring, A. Gicnapp, D. E. Hasselmann, P. Kruseman, A. Meerburg, P. Mullen, D. J. Olbers, K. Richren, W. Sell, and H. Walden, "Measurements of wind-wave growth and swell decay during the joint North Sea wave project (JONSWAP).," *Deutsche Hydrographische Zeitschrift*, vol. 12, no. Deutches Hydrographisches Institut, 1973.

[57] X. Han, B. J. Leira, S. Saevik, G. Radhakrishnan, S. Skjong, and L. T. Kyllingstad, "A framework for condition monitoring and risk-based decision support involving a vessel state observer," (Virtual, Online), Proceedings of the ASME 2021 40th International Conference on Ocean, Offshore and Arctic Engineering, 6 2021.

[58] M. S. Triantafyllou, M. Bodson, and M. Athans, "Real Time Estimation of Ship Motions Using KalmanFiltering Techniques," *IEEE Journal of Oceanic Engineering*, vol. 8, no. 1, pp. 9–20, 1983.

[59] T. Takami, U. D. Nielsen, and J. J. Jensen, "Real-time deterministic prediction of wave-induced ship responses based on short-time measurements," *Ocean Engineering*, vol. 221, no. November 2020, 2021.

[60] G. Zhang, F. Tan, and Y. Wu, "Ship Motion Attitude Prediction Based on an Adaptive Dynamic Particle Swarm Optimization Algorithm and Bidirectional LSTM Neural Network," *IEEE Access*, vol. 8, pp. 90087–90098, 2020.

[61] Q. Sun, Z. Tang, J. Gao, and G. Zhang, "Short-term ship motion attitude prediction based on LSTM and GPR," *Applied Ocean Research*, vol. 118, p. 102927, 2022.

[62] G. F. Clauss, S. Kosleck, and D. Testa, "Critical situations of vessel operations in short crested seas-forecast and decision support system," *Journal of Offshore Mechanics and Arctic Engineering*, vol. 134, 2 2012.

[63] B. S. Connell, W. M. Milewski, J. P. Rudzinsky, J. G. Kusters, C. S. Brundick, and G. Farquharson, "Development of an environmental and ship motion forecasting system," *Proceedings of the International Conference on Offshore Mechanics and Arctic Engineering - OMAE*, vol. 11, pp. 1–11, 2015.

[64] L. K. Alford, R. F. Beck, J. T. Johnson, D. Lyzenga, O. Nwogu, and A. Zundel, "A Real-time System for Forecasting Extreme Waves and Vessel Motions," (St. John's, Newfoundland, Canada), ASME 2015 34th International Conference on Ocean, Offshore and Arctic Engineering, 5 2015.

[65] P. Naaijen, D. K. Roozen, and R. H. M. Huijsmans, "Reducing Operational Risks by On-board Phase Resolved Prediction of Wave Induced Ship Motions," (Busan, South Korea), ASME 2016 35th International Conference on Ocean, Offshore and Arctic Engineering, 6 2016.

[66] K. Hessner, P. Naaijen, J. Dannenberg, and K. Reichert, "The On board Wave and Motion Estimator OWME Ocean towing of offshore structures View project Service Performance Analysis JIP View project The On board Wave and Motion Estimator OWME," tech. rep., 2010.

[67] B. Taskar, K. Hian Chua TCOMS, S. Tatsuya Akamatsu, S. Wen Yeow, and R. Niki, "Real-time ship motion prediction using artificial neural network," (Hamburg, Germany), Proceedings of the ASME 2022 41st International Conference on Ocean, Offshore and Arctic Engineering, 6 2022.

[68] M. L. Schirmann, M. D. Collette, and J. W. Gose, "Data-driven models for vessel motion prediction and the benefits of physics-based information," *Applied Ocean Research*, vol. 120, 3 2022.

[69] R. Mohammed, J. Rawashdeh, and M. Abdullah, "Machine Learning with Oversampling and Undersampling Techniques: Overview Study and Experimental Results," in *2020 11th International Conference on Information and Communication Systems, ICICS 2020*, pp. 243–248, Institute of Electrical and Electronics Engineers Inc., 4 2020.

[70] J. E. Gado, G. T. Beckham, and C. M. Payne, "Improving Enzyme Optimum Temperature Prediction with Resampling Strategies and Ensemble Learning," *Journal of Chemical Information and Modeling*, vol. 60, pp. 4098–4107, 8 2020.

[71] Miros, "Miros Wave X."

[72] A. Ross, V. Hassani, O. Selvik, and D. Fathi, "Identification of Nonlinear Manoeuvring Models for Marine Vessels Using Planar Motion Mechanism Tests," in *Proceedings of the ASME 2015 34th International Conference on Ocean, Offshore and Arctic Engineering*, (St.John's, Newfoundland, Canada), 2015.

[73] Z. Wang, C. Guedes Soares, and Z. Zou, "Optimal design of excitation signal for identification of nonlinear ship manoeuvring model," *Ocean Engineering*, vol. 196, no. December 2019, p. 106778, 2020.

[74] F. Ljungberg, J. Linder, M. Enqvist, and K. Tervo, "Dictionary-based experiment design for estimation of marine models," *Control Engineering Practice*, vol. 135, 10 2023.

[75] W. Luo and X. Li, "Measures to diminish the parameter drift in the modeling of ship manoeuvring using system identification," *Applied Ocean Research*, vol. 67, pp. 9–20, 2017.

[76] W. L. Luo and Z. J. Zou, "Parametric identification of ship maneuvering models by using support vector machines," *Journal of Ship Research*, vol. 53, pp. 19–30, 3 2009.

[77] P. Mucha and O. El Moctar, "Revisiting mathematical models for manoeuvring prediction based on modified Taylor-series expansions," *Ship Technology Research*, vol. 62, no. 2, pp. 81–96, 2015.

[78] T. Wang, G. Li, B. Wu, V. Æsøy, and H. Zhang, "Parameter identification of ship manoeuvring model under disturbance using support vector machine method," *Ships and Offshore Structures*, vol. 16, no. S1, pp. 13–21, 2021.

[79] W. Luo, C. Guedes Soares, and Z. Zou, "Parameter identification of ship maneuvering model based on support vector machines and particle swarm optimization," *Journal of Offshore Mechanics and Arctic Engineering*, vol. 138, no. 3, pp. 1–8, 2016.

[80] The International Towing Tank Conference, "ITTC-Recommended Procedures: Full Scale Measurements Manoeuvrability Full Scale Manoeuvring Trials Procedure," tech. rep., 2002.

[81] M. S. Chislett and J. Strom-Tejsen, "Planar motion mechanism tests and full-scale steering and manoeuvring predictions for a Mariner class vessel," *International Shipbuilding Progress*, vol. 12, no. 129, pp. 201–224, 1965.

[82] K. Son and K. Nomoto, "On the coupled motion of steering and rolling of a high speed container ship," *Naval Architect of Ocean Engineering*, vol. 150, no. 20, pp. 73–83, 1981.

[83] T. Perez and T. I. Fossen, "A Matlab toolbox for parametric identification of radiation-force models of ships and offshore structures," *Modeling, Identification and Control*, vol. 30, no. 1, pp. 1–15, 2009.

[84] The International Towing Tank Conference, "ITTC Quality System Manual Recommended Procedures and Guidelines: Validation of Manoeuvring Simulation Models," tech. rep., 2014.

Appendix

A

Paper I

This paper is not included due to copyright available at
https://doi.org/10.1109/JSEN.2021.3119069 and https://hdl.handle.net/11250/2827631

B

Paper II

Co-simulation-Based Pre-training of a Ship Trajectory Predictor

Motoyasu Kanazawa[1], Lars Ivar Hatledal[2(✉)], Guoyuan Li[1(✉)], and Houxiang Zhang[1(✉)]

[1] The Department of Ocean Operations and Civil Engineering, Faculty of Engineering, Norwegian University of Science and Technology, Larsgårdsvegen 2, 6009 Ålesund, Norway
{motoyasu.kanazawa,guoyuan.li,hozh}@ntnu.no

[2] The Department of ICT and Natural Sciences, Facultiy of Information Techonology and Electrical Engineering, Norwegian University of Science and Technology, Larsgårdsvegen 2, 6009 Ålesund, Norway
laht@ntnu.no

Abstract. A ship trajectory predictor plays a key role in the predictive decision making of intelligent marine transportation. For better prediction performance, the biggest technical challenge is how we incorporate prior knowledge, acquired during the design-stage experiments, into a data-driven predictor if the number of available real-world data is limited. This study proposes a new framework under co-simulation platform Vico for the development of a neural-network-based trajectory predictor with a pre-training phase. Vico enables a simplified vessel model to be constructed by merging a hull model, thruster models, and a controller using a co-simulation standard. Furthermore, it allows virtual scenarios, which describe what will happen during the simulation, to be generated in a flexible way. The fully-connected feedforward neural network is pre-trained with the generated virtual scenarios; then, its weights and biases are finetuned using a limited number of real-world datasets obtained from a target operation. In the case study, we aim to make a 30 s trajectory prediction of real-world zig-zag maneuvers of a 33.9m-length research vessel. Diverse virtual scenarios of zig-zag maneuvers are generated in Vico and used for the pre-training. The pre-trained neural network is further finetuned using a limited number of real-world data of zig-zag maneuvers. The present framework reduced the mean prediction error in the test dataset of the real-world zig-zag maneuvers by 60.8% compared to the neural network without the pre-training phase. This result indicates the validity of virtual scenario generation on the co-simulation platform for the purpose of the pre-training of trajectory predictors.

Keywords: Co-simulation · Trajectory prediction · Informed machine learning

Supported by a grant from NRF, IKTPLUSS project No. 309323 "Remote Control Center for Autonomous Ship Support" in Norway.

A. Cerone et al. (Eds.): SEFM 2021 Workshops, LNCS 13230, pp. 173–188, 2022.
https://doi.org/10.1007/978-3-031-12429-7_13

1 Introduction

Autonomous ship maneuvering hinges on better understanding of ship dynamics. A ship is regarded as a system that consists of many dynamic components, such as a hull model, thruster models, and a controller. In most cases, a simulator of each component is given by its manufacturers and project partners in the format of a black-box model. Therefore, it is of great interest to designers to construct a simplified vessel model easily by connecting black-box sub-models. *Vico*, which is a high-level co-simulation framework [3], enables a simplified vessel model to be constructed by merging a hull model, thruster models, and a controller using the co-simulation standard. Furthermore, it allows virtual scenarios, which describe what will happen during the simulation, to be generated in a flexible way. In the Intelligent Systems Lab [6] at Norwegian University of Science and Technology (NTNU) Ålesund, it has been playing an important role as a cyber testbed of research activity.

In the predictive decision making of autonomous maneuvering, a controller of an autonomous ship makes predictions of its own ship's trajectory based on the current vessel state and future command assumption. Then, a controller can evaluate and manipulate the future command assumption based on the predicted consequences. Hence, accurate motion prediction is the basis of collision avoidance algorithms [5]. According to the definition in [11], the prediction task can be explained as follows. A dataset $\mathcal{D}$ consists of a feature space $\mathcal{X}$ and a marginal probability distribution $P(X)$ where $X = \{x_1, .., x_n\} \in \mathcal{X}$. In the supervised learning, a set of pairs $\{x_i, y_i\}$ of inputs (the current vessel state, commands, and environmental disturbances) $x_i \in \mathcal{X}$ and outputs (predicted trajectories) $y_i \in \mathcal{Y}$ are given in the dataset $\mathcal{D}$. The prediction task of the own ship's trajectory $\mathcal{T}$ is defined as $\mathcal{T} = \{\mathcal{Y}, P(Y|X)\}$ where $Y = \{y_1, ..., y_n\}$. We aim to find an objective predictive function $f = P(Y|X)$ in the development of a predictor.

Ship dynamics is highly nonlinear and complex. In order to comprehend its dynamic characteristics, experiments in the ocean basin and full-scale sea trials are conducted in the design procedure. These data build a dataset $\mathcal{D}_m = \{\mathcal{X}_m, P(X_m)\}$ which is utilized for developing a white-box vessel model f_m using parameter identification algorithm [19]. Its biggest challenge is dataset of a target operation $\mathcal{D}_{\text{target}} = \{\mathcal{X}_{\text{target}}, P(X_{\text{target}})\}$ is not identical to $\mathcal{D}_m$ in most cases. Due to such factors as the shallow water effect and $P(X_{\text{target}}) \neq P(X_m)$, a white-box-model-based prediction can be inaccurate in $\mathcal{D}_{\text{target}}$ [16]. A standard idea of dealing with this problem is to sample $\mathcal{D}_{\text{train}} \subset \mathcal{D}_{\text{target}}$ and develop a black-box model trained using $\mathcal{D}_{\text{train}}$. However, it might be an optimistic expectation that we have $|\mathcal{D}_{\text{train}}|$ that is enough for the training of the black-box model since we would develop a predictor of a ship with less experience in $\mathcal{D}_{\text{target}}$. According to the definition in [18], this challenge is categorized into few-shot learning problem.

This study proposes the framework of co-simulation-based development of a ship trajectory predictor shown in Fig. 1. In a present framework, few-shot learning, which has a limited number of real-world data of a target operation, is informed of prior knowledge by many virtual operations in *Vico*. A simplified vessel model constructed at a small cost in *Vico* and real-world small data of a

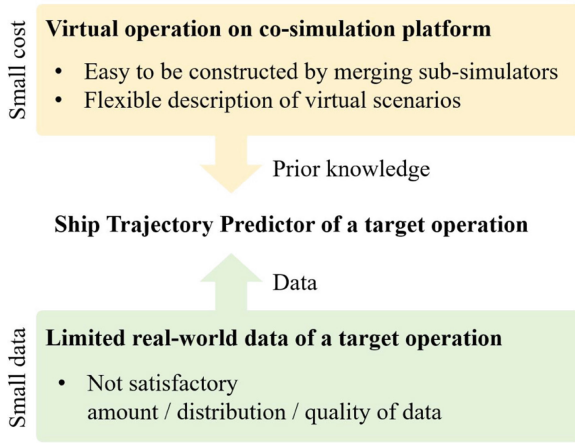


Fig. 1. Co-simulation-based development of a ship trajectory predictor.

target operation compose an accurate ship trajectory predictor only with small cost & data. One should note that we do not re-build a simplified vessel model in *Vico* by employing parameter identification algorithms and $\mathcal{D}_{\text{train}}$ since it may ruin the advantage of using the co-simulation technology, which is an easy construction of a model by merging black-box sub-models. In addition, the present approach is more attractive than a re-construction of a simplified vessel model as a formulation based on physics can not capture highly nonlinear and complex phenomena that lies behind real-world ship dynamics. We use a fully-connected Feedforward Neural Network (FNN) as an architecture of the predictor. The NN is pre-trained using diverse virtual scenarios of a target operation in *Vico* before the NN is trained by a limited number of $\mathcal{D}_{\text{train}}$ real-world data in the $\mathcal{D}_{\text{target}}$. This pre-training informs the NN of prior knowledge f_m that facilitates the training using real-world data of a target operation. A present study is a new approach that is different from any informed-machine-learning-based predictors in previous studies [9,15–17] in this field. The advantage of the proposed framework is (1) its simple structure of the predictor and (2) the NN can experience diverse virtual scenarios in the pre-training. One should note that a quantitative comparison of prediction performances between predictors is very challenging since the quantitative result could be on the case-by-case basis depending on the fidelity of prior knowledge, real-world dataset of a target operation, and the practical limitation of the implementation of the predictor. Therefore, this study focuses not on a comparison of the performances but on showing the validity of the proposed new framework. In the case study of the present framework, $\mathcal{D}_{\text{target}}$ is real-world data of zig-zag maneuvers of the R/V Gunnerus which is a 33.9m-length research vessel of NTNU. To generate virtual scenarios of zig-zag maneuvers in a pre-training phase, we develop a virtual R/V Gunnerus in *Vico* by merging a hull model provided by SINTEF Ocean, thruster models provided by thruster manufacturers, and a zig-zag controller developed by the authors.

By using the virtual scenarios, an NN-based 30 s future trajectory predictor is pre-trained without using real-world data. After the pre-training, the weights and biases are further updated using real-world data of zig-zag maneuvers $\mathcal{D}_{\text{train}}$. The contributions of the present study are summarized as follows.

- This study introduces a framework of co-simulation-based development of a ship trajectory predictor. A vessel model in co-simulation platform is easily constructed only by merging sub-models. Virtual scenarios facilitate the training of a NN-based trajectory predictor provided that limited real-world data of a target operation is available.
- The proposed framework contributes to reducing the mean prediction error by 60.9% compared to the NN-based trajectory predictor without a co-simulation-based pre-training phase.

2 Related Works

Previous studies aiming at developing accurate ship trajectory predictors are articulated in this section. Trajectory predictors are grouped into two categories; namely, white-box and black-box models. The most concise white-box model is the holonomic [20] and kinematic models [13]. They are widely used in collision avoidance algorithms because of their simple implementation, however, their prediction accuracy is much poorer than that of kinetic dynamic models due to their unrealistic assumptions. Kinetic models are categorized into response models, the Abkowitz model, the Maneuvering Modeling Group (MMG), and vectorial representations. Through experiments in the ocean basin and full-scale sea trials, hydrodynamic parameters of the kinetic models are identified using parameter identification algorithms [19]. The biggest advantage of white-box models is that they require less data to calibrate than their black-box counterparts by virtue of its formulation based on physics. On the other hand, it is a major drawback that tailored experiments take cost & effort. Black-box models exploit a large amount of onboard sensor data using Machine Learning (ML) algorithms. With the rapid development in computational resources and advanced ML algorithms, black-box models are becoming more and more popular recently in this field [7,14].

As we explained in the previous section, a general prediction problem suffers from prediction error due to $P(X_{\text{target}}) \neq P(X_m)$ and a limited number of $\mathcal{D}_{\text{train}} \subset \mathcal{D}_{\text{target}}$. With the aim of introducing prior knowledge f_m to a black-box model trained with $\mathcal{D}_{\text{train}}$, there is a large body of research integrating a mathematical vessel model into a black-box model. In [9], they utilize a reference mathematical vessel model of which dynamic characteristic is similar to the vessel that is subject to the trajectory prediction from the database. The random forest algorithm is trained so that it compensates the error in the model-based predicted acceleration. NN-based error compensation in the model-based acceleration is seen in [15]. Skulstad et al. [16] proposes a multiple-step-ahead trajectory predictor by combining a mathematical vessel model and a Long Short-Term

Memory (LSTM). An LSTM compensates 30 s North and East position errors made by the mathematical vessel model using onboard sensor measurements. Wang et al. [17] develop an NN-based data-driven calibrator that maps trajectory prediction made by a reference vessel model into that of the targeting vessel. In these previous studies, prior knowledge is integrated into the predictor in the form of a mathematical vessel model and the training is conducted using limited real-world data of a target operation. On the other hand, the present study pre-trains an NN-based predictor using diverse virtual scenarios of a target operation. This idea enables the NN to experience the diverse virtual scenarios and acquire prior knowledge before the main training rather than having a complex structure in the predictor with a black-box model and a vessel model.

3 Methodology

The methodology of the present framework is described in this section by taking an example of making trajectory prediction of zig-zag maneuvers of the R/V Gunnerus. It should be noted that the present framework works for any type of operation of any vessels as long as sub-models in *Vico* and real-world data of a target operation $\mathcal{D}_{\text{train}}$ are available.

3.1 Pre-training of a Trajectory Predictor

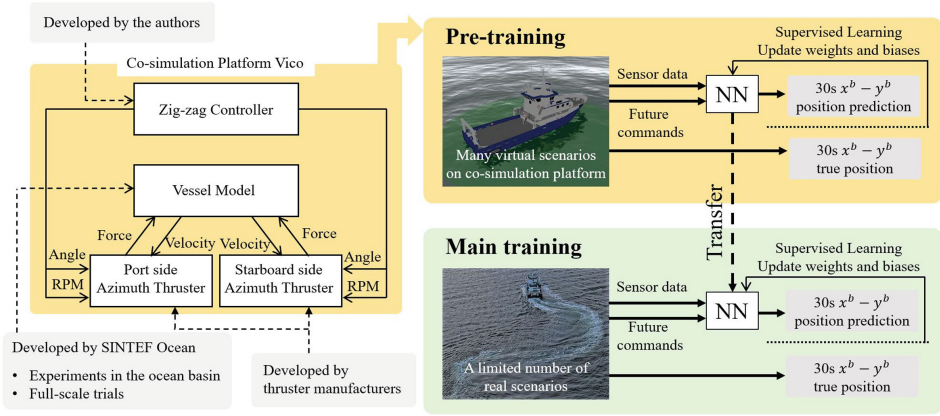


Fig. 2. The framework of co-simulation-based development of a ship trajectory predictor.

If we mix up the augmented and real-world data in one training phase, it may induce a problem of how to balance those two datasets for better prediction accuracy since the fidelity of virtual simulation might not be satisfactory in most cases of ship trajectory prediction. Therefore, the main training using real-world data

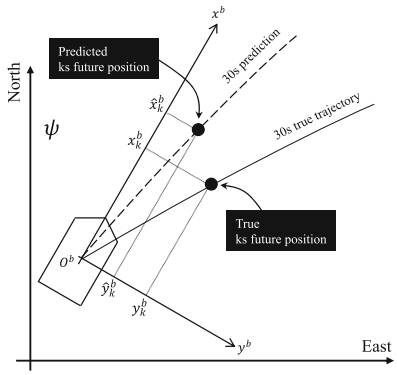


Fig. 3. The diagram of the definition of the North-East-Down frame and the body-fixed frame when making prediction.

is conducted after the pre-training using simulation data is completed. Since trajectory prediction is utilized for the predictive decision making, a vessel needs to have enough time to make a change of course and speed after making prediction. The predictor in this study makes the trajectory prediction for 30 s that is enough for making a change of course and speed of the R/V Gunnerus so that the evaluation of prediction performance is informative in the light of the practical application. Figure 2 shows the overview of the present framework through co-simulation-based pre-training. RPM in Fig. 2 is the abbreviation of the Revolution Per Minute. In a pre-training phase shown in the right top panel in Fig. 2, virtual scenarios $\mathcal{D}_{\text{pre}}$ that mimic a target operation (e.g., zig-zag maneuvers in the case study) are generated in *Vico*. Figure 3 shows the definition of the North-East-Down (NED) frame and the body-fixed $x^b - y^b - z^b$ frame when making prediction. x_k^b and y_k^b represent the true ks future x^b and y^b positions in the body-fixed frame when making prediction. ψ is the heading of the vessel to North when making prediction. $\hat{x}_k^b$ and $\hat{y}_k^b$ represent the predicted ks future x^b and y^b positions in the body-fixed frame when making prediction. An FNN f_v is pre-trained using $\mathcal{D}_{\text{pre}}$. f_v produces 30 s future trajectory prediction $[\hat{x}_1^b, ..., \hat{x}_{30}^b, \hat{y}_1^b, ..., \hat{y}_{30}^b] = f_v(\nu_0, \boldsymbol{n}, \boldsymbol{\delta})$ where ν_0 is the velocity vector when making prediction, $\boldsymbol{n} = [n_0, .., n_k, ..., n_{29}]$ is the vector of thruster revolution, n_0 is the thruster revolution when making prediction, n_k is the assumption of the thruster revolution at ks future, $\boldsymbol{\delta} = [\delta_0, .., \delta_k, ..., \delta_{29}]$ is the vector of thruster revolution, δ_0 is the thruster angle when making prediction, and δ_k is the assumption of the thruster angle at ks future. In the application of this study, the prediction $[\hat{x}_1^b, ..., \hat{x}_{30}^b, \hat{y}_1^b, ..., \hat{y}_{30}^b]$ is used for evaluating the decision making $\boldsymbol{n}$ and $\boldsymbol{\delta}$. Therefore, one should note that $\boldsymbol{n}$ and $\boldsymbol{\delta}$ are given by a controller. As illustrated in the right bottom panel in Fig. 2, the pre-trained NN is transferred to the main training phase. In the main training phase shown in the bottom panel in Fig. 2, the weights and biases of f_v are finetuned using a limited number of real-world data $\mathcal{D}_{\text{train}}$.

3.2 A Virtual Vessel Model in *VICO*

The left panel in Fig. 2 shows the structure of a virtual R/V Gunnerus in *Vico*. It consists of four components; namely, a hull model, a port-side azimuth thruster model, a starboard-side azimuth thruster, and a zig-zag controller. They are packaged into Functional Mock-up Units (FMUs; FMU1.0 for hull and thruster models and FMU2.0 for the controller). The hull model is developed by SINTEF Ocean in the SimVal project [2] through experiments in the ocean basin and full-scale sea trials. It is a 6 Degrees Of Freedom (DOF) maneuvering and seakeeping model. Port and starboard-side azimuth thruster models are provided by thruster manufactures. In order to reproduce a target operations in *Vico*, a zig-zag controller is coded in Python by the authors. It provides a pre-defined time series of thruster commands (revolution and angle) to thruster FMUs. An example of the pre-defined time series of thruster angle is shown in Fig. 5. The controller is packaged into a FMU using *PythonFMU* [4].

3.3 An FNN-Based Predictor

Many architectures of ML models have been used for ship trajectory prediction; such as Support Vector Regression (SVR) [7], LSTM [15] and fully-connected FNN [16]. In this study, we use a fully-connected FNN as an architecture of a predictor as it is one of the simplest and well-known ML models that is widely used in the context of transfer learning. The FNN-based predictor consists of the input layer, hidden layers, and output layer. The activation function is the hyperbolic tangent function for the hidden layers and the linear function for the output layer. ν_0, $\boldsymbol{n}$, and $\boldsymbol{\delta}$ are selected as input features through feature selection as explained hereinafter. The output of the predictor is a vector $[\hat{x}^b_1, ..., \hat{x}^b_{30}, \hat{y}^b_1, ..., \hat{y}^b_{30}]$ with a length of 60. The weights and biases of the FNN are updated so that it minimizes the Mean Squared Error (MSE) metric L between the true and predicted position vectors using the Adam [8] optimizer.

$$L = \frac{1}{H}\sum_{k=1}^{H}(\hat{x}^b_k - x^b_k)^2 + (\hat{y}^b_k - y^b_k)^2 \tag{1}$$

Input features are standardized with their mean and standard deviation in a training dataset in a pre-training phase. The FNN is implemented in Pytorch [12] in Python.

Feature Selection. $[x^b_1, ..., x^b_{30}, y^b_1, ..., y^b_{30}] = f(P, \nu_0, \boldsymbol{n}, \boldsymbol{\delta}) + w$ in theory of ship dynamics where P is a set of hydrodynamic and inertial parameters and w is environmental disturbances caused by wind, wave, and ocean current. In this study, we introduce following assumptions.

– A predictor is trained for a specific loading condition of a specific ship. Therefore, P is ruled out from input features of the NN.

– In most cases, a ship has no accurate measurement of waves and currents. Therefore, environmental forces due to wave and current are not modeled in the predictor.
– In $\mathcal{D}_{\text{target}}$, the effect of wind on the vessel motion is marginal. For the sake of simplicity of the validation study, it is not included in the input features of the NN.

These assumptions yield the predictor $[\hat{x}_1, ..., \hat{x}_{30}, \hat{y}_1, ..., \hat{y}_{30}] = f(\nu_0, \boldsymbol{n}, \boldsymbol{\delta})$. If accurate measurement and mathematical models of environmental disturbances in *Vico* are available, the proposed framework is valid even if the effect of environmental disturbances on ship motion is significant. One might be able to enhance prediction performances by using wind sensor and real-time acceleration data in the future work, however, the scope of this study is not to investigate the best architecture of NN models but to present the validity of present framework.

Hyperparameter Tuning. Hyperparameters are a set of parameters that need to be set prior to the training. It is well known that they have a significant impact on the performance of NNs. In a pre-training phase, the number of hidden layers *nlayers* $\in [1, 8]$, the number of units in the hidden layers *midunits* $\in [10, 500]$, learning rate *lr* $\in [1.0 \times 10^{-4}, 1.0 \times 10^{-1}]$, dropout rate in the input layer *dropin* $\in [0.0, 1.0]$, and dropout rate in the hidden layer *drophd* $\in [0.0, 1.0]$ are tuned using hyperparameter tuning framework *optuna* [1] that employs Tree-structured Parzen Estimator as an optimization algorithm. As the range of search of *lr* is wide, it is searched in the log domain. 50 *optuna* trials are conducted. We check further trials contribute to marginal improvement of the validation loss. In a main training phase, *nlayers* and *midunits* are fixed as the NN in the pre-training phase is transfered to the main training phase; then, *lr*, *dropin*, and *drophd* are tuned. When a NN is trained without a pre-training phase for comparison purposes apart from the NN trained in pre-training and main training phases, 50 *optuna* trials search an optimal set of hyperparameters *nlayers*, *midunits*, *lr*, *dropin* and *drophd*.

4 Case Study

Fig. 4. The starboard view of the R/V Gunnerus employed in the case study [10].

In a case study, we aim to make a 30 s trajectory prediction of real-world zig-zag maneuvers of the R/V Gunnerus. The starboard view of the R/V Gunnerus is shown in Fig. 4.

4.1 Pre-training

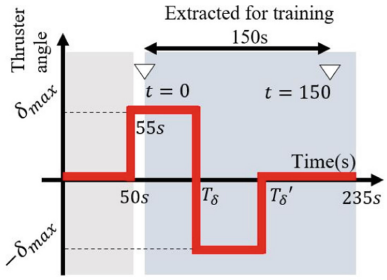


Fig. 5. Pre-defined time series of thruster angle given by a zig-zag controller.

In a pre-training phase, $n_{\text{pre}} = 300$ virtual scenarios are generated by merging a hull model, thruster models, and a zig-zag controller of the R/V Gunnerus in *Vico*. A fixed-step algorithm is used and coupling between sub-models are not considered in Vico. The virtual R/V Gunnerus is equipped with two azimuth thrusters in the port and starboard sides. These two thrusters receive the same commands of thruster angle and revolution from a zig-zag controller. Commands are simultaneously applied to thruster models since the difference between command and feedback values of the R/V Gunnerus is very small. The wave, wind, and ocean current are not applied to the virtual vessel. An example of the pre-defined time series of thruster angle is shown in Fig. 5. Each scenario is a 235 s time series. Thruster angle and revolution are set to zero before 50 s. The vessel state is reset to the initial state at 50 s. In order to avoid having impact load due to the reset, a $T_{\text{pre}} = 150\,\text{s}$ time series from 55 s ($t = 0$) to 205 s ($t = 150$) is saved in 1 Hz for the experiment with its 30 s future true positions $[x^b_1, ..., x^b_{30}, y^b_1, ..., y^b_{30}]$ and corresponding controller commands at each time step. Thruster angle is $\delta_{\max}$ until $t = T_\delta$; then, it is changed to $-\delta_{\max}$ until $t = T'_\delta = 2T_\delta$ with the maximum change rate. At $t = T'_\delta$, it is turned back to zero with the maximum change rate. Thruster revolution is set to $n_{\max}$ from 50 s to 235 s. Each scenario is parameterized by a vector of parameters $S = [\delta_{\max}, T_\delta, n_{\max}, u_{t=0}]$ where $\delta_{\max} \in [-35°, 35°]$, $T_\delta \in [t = 50\,\text{s}, t = 75\,\text{s}]$, $n_{\max} \sim \mathcal{N}(\mu = 130\,\text{RPM}, \sigma = 10\,\text{RPM})$ and the initial surge velocity $u_{t=0} \sim \mathcal{N}(\mu = 4.0\,\text{m/s}, \sigma = 1.0\,\text{m/s})$ are randomly given to each scenario. $\mathcal{N}(\mu, \sigma)$ indicates the Gaussian distribution with the mean value μ and the standard deviation σ. The probability distribution of parameters of S can be assumed based on the general understanding of a target operation $\mathcal{D}_{\text{target}}$, however, the discrepancy of the probability distribution of input features in $\mathcal{D}_{pre}$

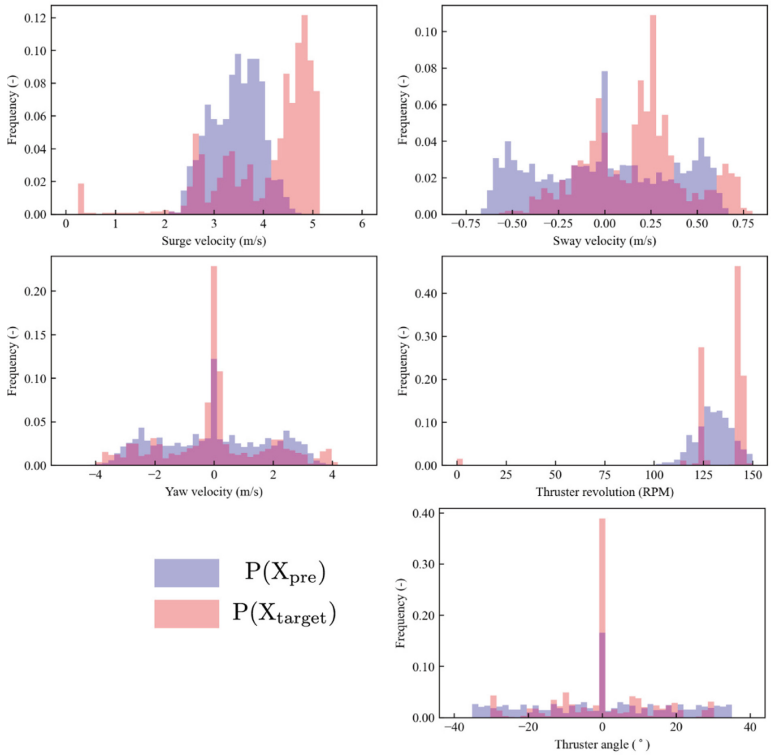


Fig. 6. Probability distributions of input features in the pre-training virtual dataset $P(X_{\text{pre}})$ and in the target real-world dataset $P(X_{\text{target}})$.

and $\mathcal{D}_{target}$ is inevitable as shown in Fig. 6. The initial North and East positions, heading, sway velocity, and yaw velocity are set to zero.

$n_{\text{pre}} = 300$ scenarios are divided into $n_{\text{train,pre}} = 192$ scenarios in the training dataset, $n_{\text{val,pre}} = 48$ scenarios in the validation dataset, and $n_{\text{test,virtual}} = 60$ scenarios in the test dataset. The test dataset is used only for checking the performance of the pre-trained predictor in the pre-training phase as shown in Fig. 9. The NN is trained only by using scenarios in the training dataset. To avoid overfitting the training dataset, the prediction performance of the trained NN in the validation dataset is monitored during the training. If the validation loss does not improve over $n_e = 200$ epochs, the training is automatically stopped; then, the best model is loaded. We checked $n_e > 200$ does not contribute to further improvement of the validation loss.

4.2 Main Training

The zig-zag maneuvers experiments of the R/V Gunnerus are conducted in November 2019 in Trondheim, Norway. Its port-side and starboard-side azimuth thrusters move simultaneously with the same commands of the thruster angle

and revolution. Its tunnel thruster is turned off during the experiments. The experiment is a 1600 s time history. We split this time history into $n_{\text{target}} = 16$ operations of which length is 100 s. As we need 30 s future positions and commands at each time step in the operation for the training and evaluation purposes, first $T_{\text{target}} = 70$ s of each operation is saved as one operation with 30 s true future positions $[x_1^b, ..., x_{30}^b, y_1^b, ..., y_{30}^b]$ at each time step. The R/V Gunnerus is equipped with 13 onboard sensors sampling ship motion in real time. During the experiment, positions in the NED frame (North, East, and heading) and velocities in the body-fixed frame (surge, sway, and yaw speed) are saved in 1 Hz. Equation (2) converts positions in the NED frame into $[x_1^b, ..., x_{30}^b, y_1^b, ..., y_{30}^b]$ in the body-fixed frame:

$$\begin{pmatrix} x_k^b \\ y_k^b \end{pmatrix} = \begin{pmatrix} \cos\psi & \sin\psi \\ -\sin\psi & \cos\psi \end{pmatrix} \begin{pmatrix} N_k - N_0 \\ E_k - E_0 \end{pmatrix} \tag{2}$$

where N_k and E_k are the true ks future North and East positions in the NED frame. N_0 and E_0 are North and East positions when making prediction in the NED frame. At each time step, future command assumptions $\boldsymbol{n}$ and $\boldsymbol{\delta}$ are given by the dataset as we examine the prediction performance provided that they are assumed by a controller. As we assume limited real-world data of a target operation are available, we use only $n_{\text{trainval,target}} = 12$ operations in the main training and keep the other $n_{\text{test,target}} = 4$ operations, that are used only for the evaluation of the prediction performance, untouched in the training process. $n_{\text{trainval,target}} = 12$ operations are divided into $n_{\text{train,target}} = 9$ operations in the training dataset and $n_{\text{val,target}} = 3$ operations in the validation dataset. The NN is trained only by using the training dataset and its performance in the validation dataset is monitored using the validation dataset. If the validation loss does not improve over $n_e = 200$ epochs, the training is automatically stopped; then, the best model is loaded as explained in the previous subsection. By switching the validation dataset four times, four independent NNs are trained (cross-validation). The final prediction to the untouched test dataset is the average of predictions made by these four NNs. In order to examine the contribution of the pre-training, three different strategies of training are investigated as follows.

(A) Without Pre-training. The training of this predictor is conducted without a pre-training phase. Virtual scenarios generated in *Vico* are kept untouched and only limited real-world data of a target operation $n_{\text{train,target}}$ is used in the training. It provides a baseline of the comparison study.

(B) Without Finetuning. This predictor is pre-trained with $n_{\text{train,pre}}$ virtual scenarios, however, the main training is not performed. The prediction performance of this predictor in $\mathcal{D}_{\text{target}}$ reveals the effect of the discrepancy between $\mathcal{D}_{\text{pre}}$ and $\mathcal{D}_{\text{target}}$.

(C) Present Study. This is a predictor that is trained in the manner of the present framework. The main training of this predictor is carried out with real-world data $n_{\text{train,target}}$ after the pre-training with virtual scenarios $n_{\text{train,pre}}$.

4.3 Evaluation Metric

This study introduces an evaluation metric $\overline{S}_{ijk,\text{target}}$ that indicates the mean prediction error over the 30 s prediction horizon in the test dataset of a main training phase.

$$\overline{S}_{ijk,\text{target}} = \frac{1}{n_{\text{test,target}} T_{\text{target}} H} \sum_{i=0}^{n_{\text{test,target}}} \sum_{j=0}^{T_{\text{target}}-1} \sum_{k=1}^{H} S_{ijk,\text{target}} \tag{3}$$

where $S_{ijk,\text{target}}$ is the distance between the predicted and true positions at ks prediction horizon of time step js prediction of ith scenario in the test dataset of a main training phase. In order to examine the prediction performance in a pre-training phase, we use an evaluation metric $\overline{S}_{ijk,\text{pre}}$ as follows.

$$\overline{S}_{ijk,\text{pre}} = \frac{1}{n_{\text{test,pre}} T_{\text{pre}} H} \sum_{i=0}^{n_{\text{test,pre}}} \sum_{j=0}^{T_{\text{pre}}-1} \sum_{k=1}^{H} S_{ijk,\text{pre}} \tag{4}$$

where $S_{ijk,\text{pre}}$ is the distance between the predicted and true positions at ks prediction horizon of time step js prediction of ith scenario in the test dataset in a pre-training phase.

4.4 Results

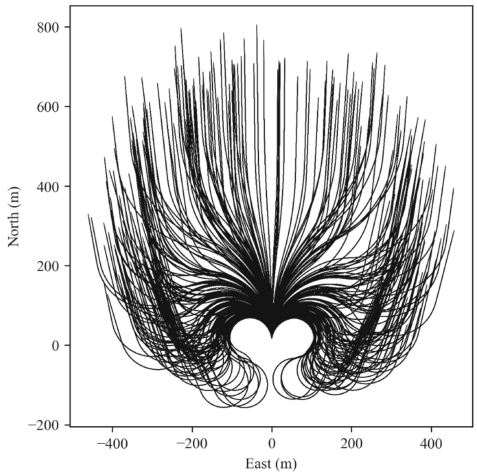


Fig. 7. Trajectories of virtual scenarios in $\mathcal{D}_{\text{pre}}$.

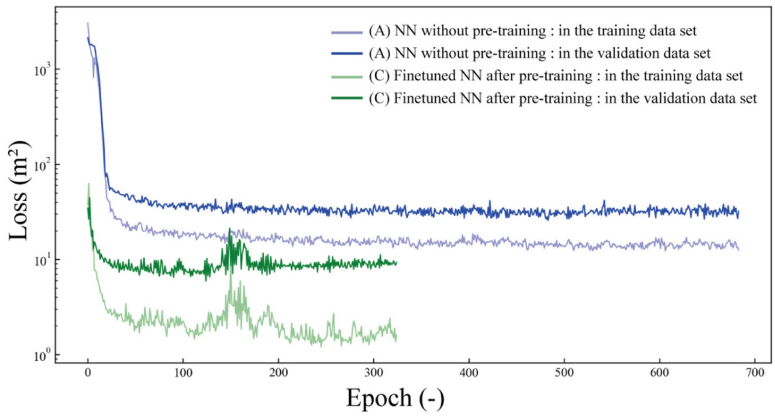


Fig. 8. Histories of the training and validation loss of the first fold of (A) the NN without pre-training and (C) the finetuned NN with pre-training. The training is terminated when the validation loss does not improve over $n_e = 200$ epochs. (Color figure online)

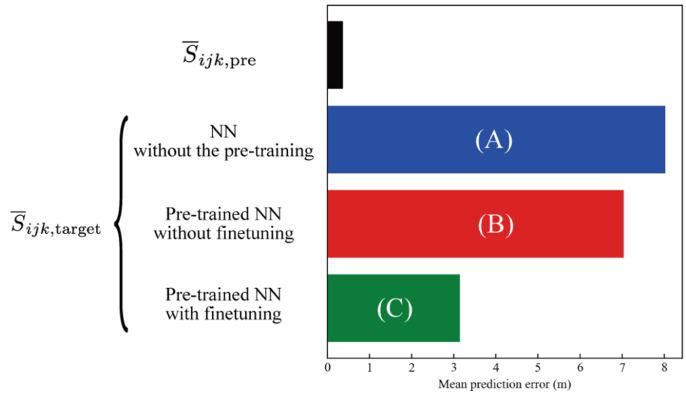


Fig. 9. Mean prediction errors in the test dataset.

Trajectories of $n_{\mathrm{pre}} = 300$ virtual scenarios generated in *Vico* are shown in Fig. 7. It is seen that diverse scenarios are generated thanks to the setting of scenario generation in the pre-training phase. Through the *optuna* hyperparameter optimization in the pre-training phase, $nlayer = 1$ and $midunits = 420$ are selected. Apart from the pre-training phase, (A) without pre-training is trained only by using real-world data $n_{\mathrm{trainval,\ target}}$. $nlayer = 1$ and $midunits = 300$ are selected through *optuna* hyperparameter optimization. After the pre-training phase, the main training is conducted. With a set of optimized hyperparameters, Fig. 8 shows histories of the training and validation losses of the first fold of the cross-validation of (A: blue lines) the NN without the pre-training phase and (C: green lines) the present study with the pre-training phase. The vertical axis displays loss values along a logarithmic scale. It should be noted that the training is

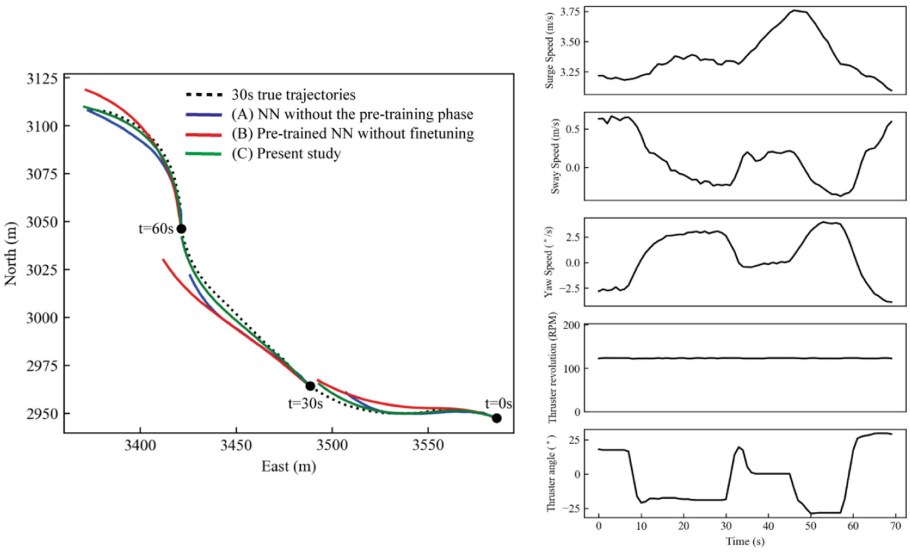


Fig. 10. (left) Snapshots of 30 s prediction at $t = 0$ s, $t = 30$ s, and $t = 60$ s of one scenario in the test dataset (right) Time histories of vessel state and commands.

automatically terminated if the validation loss does not improve over $n_e = 200$ epochs. Since (A) is not pre-trained by virtual scenarios, its training starts with notably higher loss values than that of (C). Accordingly, the validation loss of (A) ends up with higher values than that of (C). This result indicates the pre-training phase based on virtual scenarios facilitates not only the initial stage of the training but also the overall training efficiency in the main training.

Figure 9 shows the mean prediction error in the test dataset in the pre-training phase $\overline{S}_{ijk,\mathrm{pre}}$ and that in the main training phase $\overline{S}_{ijk,\mathrm{target}}$. By comparing the bars in the top and the third from the top in Fig. 9, one can see that the prediction performance of the pre-trained NN deteriorates much in $\mathcal{D}_{\mathrm{target}}$ if it is not finetuned in the main training phase due to the difference between $\mathcal{D}_{\mathrm{pre}}$ and $\mathcal{D}_{\mathrm{target}}$. The bar (A) in the second from the top in Fig. 9 reveals that the training without the pre-training phase produces the largest prediction error in (A), (B), and (C). (C) trained in the present framework with pre-training and main training phases reduces prediction error notably; by 60.8% compared to (A) without pre-training.

The left panel of Fig. 10 shows snapshots of 30 s prediction at $t = 0$ s, $t = 30$ s, and $t = 60$ s of one operation in the test dataset of real-world data in the main training phase. The right panel of Fig. 10 shows time histories of vessel state and commands of the operation. (A) the NN without the pre-training phase deviates significantly from the true trajectories at $t = 0$ s and $t = 30$ s. (B) the NN without finetuning after the pre-training phase succeeded at capturing the trend of the 30 s true trajectory at $t = 0$ s, $t = 30$ s, and $t = 60$ s in the short prediction horizon, however, it ends up with the large prediction error in the

distant prediction horizon. (C) present study with the pre-training and main training phases traces the true trajectories more accurately than (A) and (B) at $t = 0\,\mathrm{s}$, $t = 30\,\mathrm{s}$, and $t = 60\,\mathrm{s}$.

5 Conclusion

This study proposed a new framework for co-simulation-based development of a ship trajectory predictor provided that limited real-world dataset of a target operation is available. We integrated prior knowledge, which is virtual scenarios generated by a simplified vessel model in co-simulation platform *Vico*, into training of the neural-network-based trajectory predictor. The neural network is pre-trained using many virtual scenarios generated in *Vico* before it is finetuned using limited real-world dataset of a target operation. In the case study, we employed real-world operations of zig-zag maneuvers of a 33.9m-length research vessel. For pre-training, 300 virtual scenarios of zig-zag maneuvers were generated in *Vico* only by merging sub-models provided by different project partners. The pre-trained neural network was further finetuned using 12 real-world operations of zig-zag maneuvers. The present framework reduces the mean prediction error by 60.8% in the test dataset of real-work operation compared to the neural network without pre-training. Hence, the present framework enables a ship trajectory predictor to be constructed only by using a simplified vessel model in co-simulation platform at a small cost and limited real-world data of a target operation.

References

1. Akiba, T., Sano, S., Yanase, T., Ohta, T., Koyama, M.: Optuna: a next-generation hyperparameter optimization framework. In: KDD 2019: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 2623–2631. Association for Computing Machinery, Anchorage (2019). https://doi.org/10.1145/3292500.3330701
2. Hassani, V., Fathi, D., Ross, A., Sprenger, F., Selvik, Ø., Berg, T.E.: Time domain simulation model for research vessel Gunnerus. In: Proceedings of the International Conference on Offshore Mechanics and Arctic Engineering - OMAE, St. John's, Newfoundland, Canada, vol. 7 (2015). https://doi.org/10.1115/OMAE201541786
3. Hatledal, L.I., Chu, Y., Styve, A., Zhang, H.: Vico: an entity-component-system based co-simulation framework. Simul. Modell. Pract. Theory **108**(September 2020), 102243 (2021). https://doi.org/10.1016/j.simpat.2020.102243
4. Hatledal, L.I., Zhang, H., Collonval, F.: Enabling python driven co-simulation models with PythonFMU. In: 34th International ECMS Conference on Modelling and Simulation (2020). https://doi.org/10.7148/2020-0235
5. Huang, Y., Chen, L., Chen, P., Negenborn, R.R., Gelder, P.H.A.J.M.V.: Ship collision avoidance methods: state-of-the-art. Saf. Sci. **121**(April 2019), 451–473 (2020). https://doi.org/10.1016/j.ssci.2019.09.018
6. Intelligent Systems Lab @ NTNU Aalesund: Intelligent Systems Lab webpage. http://org.ntnu.no/intelligentsystemslab/

7. Kawan, B., Wang, H., Li, G., Chhantyal, K.: Data-driven modeling of ship motion prediction based on support vector regression. In: Proceedings of the 58th Conference on Simulation and Modelling (SIMS 58) Reykjavik, Iceland, 25th–27th September 2017, vol. 138, pp. 350–354 (2017). https://doi.org/10.3384/ecp17138350
8. Kingma, D.P., Ba, J.L.: Adam: a method for stochastic optimization. In: 3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings, pp. 1–15 (2015)
9. Mei, B., Sun, L., Shi, G.: White-black-box hybrid model identification based on RM-RF for ship maneuvering. IEEE Access **7**, 57691–57705 (2019). https://doi.org/10.1109/ACCESS.2019.2914120
10. Norges teknisk-naturvitenskapelige universitet: Research vessel: R/V Gunnerus. https://www.ntnu.edu/oceans/gunnerus
11. Panigrahi, S., Nanda, A., Swarnkar, T.: A survey on transfer learning. Smart Innov. Syst. Technol. **194**, 781–789 (2021). https://doi.org/10.1007/978-981-15-5971-6_83
12. Paszke, A., et al.: PyTorch: an imperative style, high-performance deep learning library. In: NeurIPS (2019)
13. Perera, L.P.: Navigation vector based ship maneuvering prediction. Ocean Eng. **138**(November 2016), 151–160 (2017). https://doi.org/10.1016/j.oceaneng.2017.04.017
14. Rong, H., Teixeira, A.P., Guedes Soares, C.: Ship trajectory uncertainty prediction based on a Gaussian Process model. Ocean Eng. **182**(December 2018), 499–511 (2019). https://doi.org/10.1016/j.oceaneng.2019.04.024
15. Skulstad, R., Li, G., Fossen, T.I., Vik, B., Zhang, H.: A hybrid approach to motion prediction for ship docking - integration of a neural network model into the ship dynamic model. IEEE Trans. Instr. Meas. **70**, 1–11 (2021). https://doi.org/10.1109/TIM.2020.3018568
16. Skulstad, R., Li, G., Fossen, T.I., Wang, T., Zhang, H.: A co-operative hybrid model for ship motion prediction. Model. Ident. Control: Norwegian Res. Bul. **42**(1), 17–26 (2021). https://doi.org/10.4173/mic.2021.1.2
17. Wang, T., Li, G., Hatledal, L.I., Skulstad, R., Aesoy, V., Zhang, H.: Incorporating approximate dynamics into data-driven calibrator: a representative model for ship maneuvering prediction. IEEE Trans. Ind. Inform. 1 (2021). https://doi.org/10.1109/tii.2021.3088404
18. Wang, Y., Yao, Q., Kwok, J.T., Ni, L.M.: Generalizing from a few examples: a survey on few-shot learning. ACM Comput. Surv. **53**(3), 1–34 (2020). https://doi.org/10.1145/3386252
19. Wang, Z., Zou, Z., Soares, C.G.: Identification of ship manoeuvring motion based on nu-support vector machine. Ocean Eng. **183**(January), 270–281 (2019). https://doi.org/10.1016/j.oceaneng.2019.04.085
20. Zhao, L., Roh, M.I.: COLREGs-compliant multiship collision avoidance based on deep reinforcement learning. Ocean Eng. **191**(May), 106436 (2019). https://doi.org/10.1016/j.oceaneng.2019.106436

C

Paper III

This paper is not included due to copyright
available at https://doi.org/10.1109/JSEN.2022.3171036

D

Paper IV

Ocean Engineering 266 (2022) 112998

Contents lists available at ScienceDirect

Ocean Engineering

journal homepage: www.elsevier.com/locate/oceaneng

Knowledge and data in cooperative modeling: Case studies on ship trajectory prediction

Motoyasu Kanazawa *, Tongtong Wang, Robert Skulstad, Guoyuan Li, Houxiang Zhang

The Department of Ocean Operations and Civil Engineering, Norwegian University of Science and Technology, 6009 Ålesund, Norway

ARTICLE INFO

Keywords:
Intelligent transportation systems
Ship automation
Trajectory prediction
Hybrid modeling
Model identification

ABSTRACT

A ship automation will be a key to the future maritime. In particular, ship dynamic models play an integral role. However, it is challenging to develop an accurate model readily. Recent studies proposed a physics-data cooperative model that predicts a future trajectory by compensating for position errors made by the physics-based model by using a machine learning model, which learns such a multiple-step-ahead compensation based on onboard sensor measurements. It seems to be promising to reduce effort in model development by exploiting observation data while having physics knowledge and a stable foundation in the model. However, it has been an open question "how much does the cooperative model benefit from physics knowledge and observation data?". We tackled this key question experimentally. To investigate the benefit of the physics-based model and the data amount, by changing the accuracy of the physics-based model and the size of observation dataset, simulation and full-scale experiments were conducted. Results show that the accuracy of the physics-based model and the data amount were complementary to each other to some extent. A wide range of physics-based models worked as prior knowledge, however, too inaccurate models disturbed the training.

1. Introduction

Recently, ship autonomy has gained an increasing attention from the research and industrial communities for ensuring operational safety and efficiency in the busy marine traffic, such as narrow channels and ports. They are expected to open new vistas in supporting or even substituting human onboard decision making to avoid human errors and make more efficient decisions (Norwegian Shipowners Association, 2019). Although we have seen the rapid success of the autonomous field robots and cars, that for autonomous ships remains topical. One of the reasons is the fact that an autonomous ship is a more comprehensive system composed of versatile marine robotics, automation, and sensing technologies.

In particular, it is of great importance to have a good situation awareness understanding what is happening now and will happen in the future surrounding a ship (e.g., Xiao et al. (2020) and Zhang et al. (2022)). A ship is a dynamic system with poor maneuverability, thereby, poor situation awareness may easily lead to fatal consequences, such as colliding with obstacles or stranding. In this context, for decades, researchers have been devoting their research effort to building an accurate ship dynamic model so that it predicts a future trajectory used for the early warning & prediction of the future collision risk. Fig. 1 illustrates the relationship between the ship dynamic model identification and the early warning & prediction in the versatile technologies for the ship autonomy in the future maritime transport.

Research communities have grouped approaches of ship dynamics modeling into three categories; namely, model-based, data-driven, and cooperative approaches. Model-based approaches formulate a linear regression model based on the understanding of physics. This study refers to such models as physics-based models. Model's parameters are identified by employing numerical simulations and model/full-scale experimental data (e.g., Wang et al. (2019)). The biggest advantage lies in the fact that they can be easily calibrated with a small dataset finding an optimal function parsimoniously in a parametric manner. Moreover, we can easily check the model's validity by looking into identified parameters. They have been playing a dominant role in applications by providing a stable and reliable foundation of the understanding of ship dynamics. However, the practice of parameter identification is rather sensitive revolving around a good-quality, sufficient, and balanced dataset in addition to the explicit understanding of real-world phenomena. Thereby, mostly, it ends up with poor performance although it takes prohibitive time & cost to be built with physically reasonable parameters. On the other hand, data-driven approaches offer parsimonious non-parametric models to achieve better performance by finding patterns in the dataset without depending on scientific

* Corresponding author.
E-mail address: motoyasu.kanazawa@ntnu.no (M. Kanazawa).

https://doi.org/10.1016/j.oceaneng.2022.112998
Received 30 August 2022; Received in revised form 10 October 2022; Accepted 22 October 2022
Available online 8 November 2022

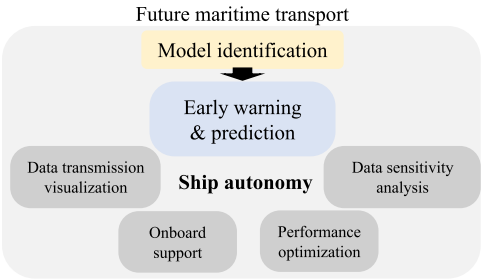


Fig. 1. A schematic overview of ship dynamic model identification and the early warning & prediction of the future collision risk in versatile technologies for ship autonomy in future maritime transport.

knowledge, although it may require more data to be calibrated than the physics-based model. Since the wake of the so-called third wave of artificial intelligence since the 1990s, data-driven approaches have been applied to challenging tasks in the maritime domain (e.g., the evaluation of ship pollutant emissions (Xiao et al., 2022) and the ship detection from videos (Chen et al., 2021)). Ship dynamics modeling is no exception to this trend (e.g., Kawan et al. (2017) and Schirmann et al. (2022)). However, pure data-driven models have no scientific interpretability. It becomes more of an issue in the application since the maritime industry is highly conservative in safety-critical systems. In addition, it is not a wise step for data-driven models to discard our domain knowledge packaged into the physics-based model.

A cooperative approach combines model-based and data-driven approaches. Recently, the maritime applications are not exceptions in a trend of making a synergy of scientific knowledge and data (e.g., Fonseca and Gaspar (2021)). Skulstad et al. (2021a) and Wang et al. (2021) presented breakthrough ideas that compensate/map trajectories made by the available physics-based model into true trajectories in a data-driven manner. Such data-driven geometrical compensation/mapping achieved a good performance while keeping the stability of the physics-based model untouched as a stable foundation of the model. Moreover, in their approaches, we can clearly distinguish the contributions of the physics-based and data-driven models on the prediction performance as they play different roles in the model, thus having good interpretability and maintainability of the model with a good performance. It seems to be promising to overcome time & cost challenges the maritime industry faces since it would lower rigorous hurdles of model-based or data-driven approaches by combining two approaches.

An open question about such cooperative approaches is "how much does the cooperative model benefit from physics knowledge and observation data?" to achieve a good performance in a physics-data cooperative way (referred to as the cooperative performance, hereinafter). In not only the academic but also the industrial views, this question is important from the two perspectives. On the one hand, the accuracy of the available physics-based model is diverse. It could be degraded, for instance, due to the poor conduct of the parameter identification, parameters identified in a compromised manner (e.g., copy & paste parameters of similar ships), low-fidelity actuator models, and being tuned to the other operations. On the other hand, available data are mostly limited. Real-world ship maneuvers are required for the data collection, however, it is money- and time-consuming. Thereby, a better understanding of the impact of these two components on the cooperative performance is of great interest to our industrial partners. To the best of the authors' knowledge, so far, this open question has not been addressed in any literature, albeit its importance in industrial applications. In this study, we validate a cooperative framework, which builds an accurate ship dynamic model by combining a compromised physics-based model and limited observation data.

To offer one solution to the open question "how much does the cooperative model benefit from physics knowledge and observation data?", this study conducted experimental investigations, which are divided into two parts. First, simulation experiments enabled us to investigate the impact of the accuracy of the physics-based model and the data amount on the cooperative performance. Second, in the full-scale experiment where only limited observation data are available, we further explore the reasonable range of the physics-based model's accuracy on the cooperative performance. The full-scale experiment was conducted by the 33.9m-length research vessel Gunnerus. The results showed that we could achieve a good performance by using the combination of the compromised physics-based model and a small dataset. The cooperative performance was equivalent to the performance of the accurate physics-based model, which takes much more time & cost to be built. Contributions of this study are summarized as follows:

- It was found that the balance of the accuracy of the physics-based model and the data amount was key to achieve a good performance of the physics-data cooperative model rather than relying on either of them. In addition, the full-scale experiment presented the validity of building a cooperative model with a compromised physics-based model and a small observation dataset. These findings make the cooperative model more promising for reducing effort dedicated for the model development by using physics knowledge and observation data.
- Although a wide range of physics-based models successfully facilitated the model identification, however, it disturbed the training if it was too inaccurate. This finding highlights the importance of technologies that develop a simplified physics-based model readily without compromising its performance drastically.

Hereinafter, this paper unfolds as follows. Section 2 illustrates related works aiming at the synergy of the scientific knowledge and data in different applications. In Section 3, we explain the cooperative ship dynamic model employed in this study. An experimental study in the simulation environment is presented in Section 4. A full-scale experiment is illustrated in Section 5. Conclusions are given in Section 7.

2. Related works

In many applications, it is seen to leverage scientific knowledge and data for better performance and reliability. In Karpatne et al. (2016), Karpatne et al. named such approaches Theory-Guided Data Science (TGDS), and grouped diverse approaches into five categories; namely, theory-guided design of data science models, theory-guided learning of data science models, theory-guided refinement of data science outputs, hybrid models of data science and theory, and augmenting theory-based models using data science. Besides TGDS, different terminologies (e.g., transfer learning (Panigrahi et al., 2021), physics-informed Machine Learning (ML) (Karniadakis et al., 2021), informed ML (Vonrueden et al., 2021), and gray-box/semi-parametric modeling (von Stosch et al., 2014)) fully/partly cover ideas of the cooperative approach in the field of ship dynamics. There is no domain-agnostic "best practice" of the cooperative approach, thereby, it is necessary to carry out a domain-specific investigation to achieve the good harmony of scientific knowledge and data.

In this study, we assume we have a physics-based model and a new dataset of the ship maneuver. The physics-based model could be derived in the compromised manner and the new dataset is not satisfactory in terms of its amount, quality, or distribution. In such settings, the most straightforward and classic way to model ship dynamics is to re-identify parameters of the physics-based model so that it performs well in the prepared dataset by using ML algorithms (e.g., support vector machine (Wang et al., 2019; Luo and Li, 2017), a Bayesian approach (Xue et al., 2020)). This approach belongs to "theory-guided

(a) Simulation experiment

(b) Full-scale experiment

Fig. 2. Snapshots of (a) simulation experiments in Section 4 and (b) full-scale experiments in Section 5. (b) was taken on November 21st, 2019 in the west coast of Norway.

design of data science models" in the TGDS's categorization. If the identification succeeds, it provides a highly interpretable model. However, it has been a challenging task since our understanding of ship dynamics mostly does not fully capture real-world phenomena. This challenge brought us to use non-parametric models in this domain. Since the maritime industry highly values the reliability of physics-based models, previous research in this field has devoted its effort to building a cooperative architecture of the physics-based and data-driven models. It corresponds to "hybrid models of data science and theory" in the TDGS's categorization. Ven et al. employed a neural network for representing damping terms of the physics-based model (van de Ven et al., 2007). In Skulstad et al. (2021b) and Kanazawa et al. (2022), neural networks were used for compensating for the single-step-ahead prediction error made by physics-based models. Mei et al. (2019) employed a random forest to map the estimation of the acceleration made by the similar ship's dynamic model into that of the targeting ship by using a dataset of the targeting ship. Those approaches directly intervene in the performance of the physics-based model by data-driven models, thus making one unified trajectory in the physics-data cooperative manner. Their cooperative approaches are efficient thanks to their simple architecture, however, their stability in the numerical iteration is hardly validated. Moreover, once trajectories are generated, it is impossible to isolate the contribution of the physics-based and data-driven models from the generated trajectory. Thereby, in practice, they can be used only when we have a relatively-accurate physics-based model due to reliability reasons.

On the other hand, Skulstad et al. (2021a) trained a neural network with onboard sensor measurements so that it compensates for the multiple-step-ahead position error made by the physics-based model. In Wang et al. (2021), Wang et al. proposed a data-driven model that maps the future position calculated by the similar ship's dynamic model into that of the targeting ship. In their approaches, the roles of the physics-based and data-driven models are clearly distinguished, thus contributing to better interpretability and maintainability of the cooperative architecture. Moreover, in their approaches, the physics-based model serves not only as prior knowledge of ship dynamics but also as a stable foundation of the prediction.

3. Cooperative ship model

As explained in Section 2, in the field of ship dynamics, previous studies have presented different types of cooperative models combining physics-based and data-driven models. In this study, we employ a geometry-based cooperative model, that makes a data-driven compensation for multiple-step-ahead position errors made by the physics-based model, based on the idea presented in Skulstad et al. (2021a) and Kanazawa et al. (2021).

3.1. Overview

In the experiments of this study, we employed a cooperative model of ship dynamics as shown in Fig. 3. The cooperative model is composed of a physics-based model highlighted in orange and a data-driven compensator highlighted in green. The physics-based model makes Ts prediction of a future trajectory based on the initial state of the ship, environmental disturbances, and commands to actuators. On the other hand, based on onboard measurement data, the data-driven compensator compensates for errors in the position made by the physics-based model, Thus, a multiple-step-ahead position prediction vector made by the physics-based model is calibrated in a data-driven manner. By adding outputs of both of them, the cooperative model makes Ts prediction of the future trajectory in a data-driven manner while having a stable and reliable model-based prediction made by the physics-based model as its foundation. Details of the physics-based model and data-driven compensator will be explained hereinafter.

3.2. Physics-based model

In the maneuvering theory of ship dynamics, the ship kinematics is expressed as:

$$\dot{\boldsymbol{\eta}} = \boldsymbol{R}(\psi)\boldsymbol{\nu} \tag{1}$$

where $\boldsymbol{\eta}$ is the vector of the ship's positions in the inertial coordinate, $\boldsymbol{R}$ is the rotation matrix between the inertial and body-fixed coordinate, ψ is the ship's heading, and $\boldsymbol{\nu}$ is the vector of the ship's velocities in the body-fixed coordinate. We define $\boldsymbol{R}$ as:

$$\boldsymbol{R}(\psi) = \begin{bmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2}$$

The ship kinetics is expressed as:

$$\boldsymbol{M_{RB}}\dot{\boldsymbol{\nu}} + \boldsymbol{M_A}\dot{\boldsymbol{\nu}}_{\boldsymbol{r}} + \boldsymbol{C_{RB}}(\boldsymbol{\nu})\boldsymbol{\nu} + \boldsymbol{C_A}(\boldsymbol{\nu_r})\boldsymbol{\nu_r} + \boldsymbol{D}(\boldsymbol{\nu_r}) = \boldsymbol{q} \tag{3}$$

$$\boldsymbol{q} = \boldsymbol{q}_{\text{wind}} + \boldsymbol{q}_{\text{wave}} + \boldsymbol{q}_{\text{thr}} \tag{4}$$

where $\boldsymbol{M_{RB}}$ is the rigid-body mass matrix, $\boldsymbol{M_A}$ is the added-mass matrix, $\boldsymbol{C_{RB}}(\boldsymbol{\nu})$ is the rigid-body coriolis-centripetal matrix, $\boldsymbol{C_A}(\boldsymbol{\nu_r})$ is the added-mass coriolis-centripetal matrix, $\boldsymbol{D}(\boldsymbol{\nu_r})$ is the damping matrix, $\boldsymbol{\nu_r} = \boldsymbol{\nu} - \boldsymbol{\nu_c}$ is the relative velocity vector, $\boldsymbol{\nu_c}$ is the current velocity vector, $\boldsymbol{q}_{\text{wind}}$ is the wind-force vector, $\boldsymbol{q}_{\text{wave}}$ is the wave-force vector, and $\boldsymbol{q}_{\text{thr}}$ is the thruster force vector. In this study, we assume the effects of the ocean current and waves on the ship motion are marginal due to the limitation that ships are not equipped with sensors measuring ocean currents and waves in real time. This assumption is acceptable under mild environmental disturbances. Thereby, this study introduces $\boldsymbol{\nu_r} = \boldsymbol{\nu}$ and $\boldsymbol{q}_{\text{wave}} = \boldsymbol{0}$. The wind-force model is constructed by dedicated numerical simulation and experiments as:

$$\boldsymbol{q}_{\text{wind}} = \frac{1}{2}\rho_a V_{rw}^2 \begin{bmatrix} C_X(\gamma_{rw})A_{FW} \\ C_Y(\gamma_{rw})A_{LW} \\ C_N(\gamma_{rw})A_{LW}L_{oa} \end{bmatrix} \tag{5}$$

where C_X, C_Y, and C_Z are the wind coefficients identified for the surge, sway, and yaw directions. A_{FW} and A_{LW} are the frontal and

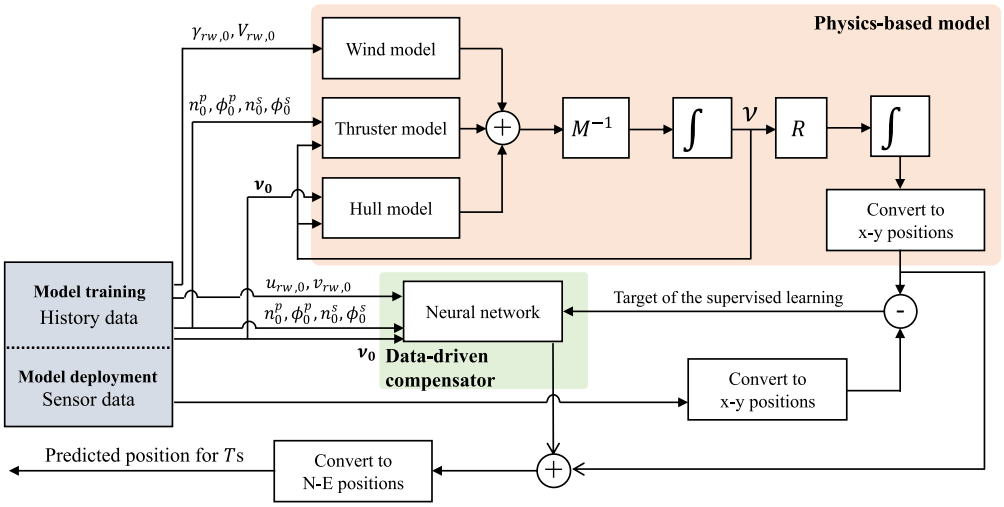


Fig. 3. An overview of the cooperative model. Sensor data (current ship's states, thruster command values, and wind information) are given to the physics-based model and data-driven compensator. The physics-based model is a 3DOF maneuvering model outputting a trajectory prediction. The data-driven compensator, of which input–output relationship is shown in (13), compensates for multiple-step-ahead position errors made by the physics-based model.

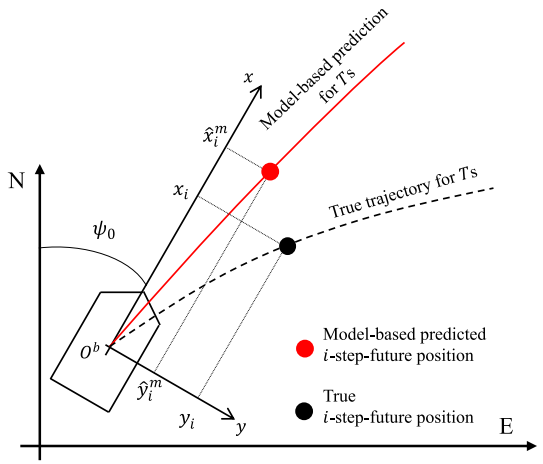


Fig. 4. A schematic relationship between the body-fixed coordinate when making a prediction and the inertial coordinate. The data-driven compensator compensates for position errors made by the physics-based model in the body-fixed coordinate when making a prediction.

lateral projected areas, respectively. L_{oa} represents the ship length. The relative wind velocity V_{rw} and direction γ_{rw} are given as:

$$V_{rw} = \sqrt{u_{rw}^2 + v_{rw}^2} \tag{6}$$

$$\gamma_{rw} = -\text{atan2}(v_{rw}, u_{rw}) \tag{7}$$

where:

$$u_{rw} = u - V_w \cos(\beta_w - \psi) \tag{8}$$

$$v_{rw} = v - V_w \sin(\beta_w - \psi) \tag{9}$$

V_w and β_w are the true wind velocity and direction in the inertial coordinate. The thruster-force vector $\boldsymbol{q}_{\text{thr}}$ is calculated based on thruster commands and $\boldsymbol{\nu}$ by the mathematical model f_{thr} provided by manufacturers of thrusters. In this study, the ship is equipped with two azimuth thrusters and one bow thruster. The bow thruster was turned off in the experiments. Namely:

$$\boldsymbol{q}_{\text{thr}} = f_{thr}(\boldsymbol{\nu}, n^p, \delta^p, n^s, \delta^s) \tag{10}$$

where n^p and n^s represent thruster revolutions of the port- and starboard-side azimuth thrusters. δ^p and δ^s are thruster angles of the port- and starboard-side thrusters. Hence, by using wind, thruster, and hull models, total forces and moment acting on the hull are calculated. By multiplying $\boldsymbol{M}^{-1} = (\boldsymbol{M_{RB}} + \boldsymbol{M_A})^{-1}$, the acceleration vector is estimated. We numerically integrate the estimated acceleration vector into the velocity vector in the body-fixed coordinate. The numerical integration of this velocity vector over the prediction horizon yields the model-based predicted trajectory $[\hat{N}_1^m, \ldots, \hat{N}_i^m, \ldots, \hat{N}_{n_T}^m, \hat{E}_1^m, \ldots, \hat{E}_i^m, \ldots, \hat{E}_{n_T}^m]$ in the future, where $\hat{N}_i^m$ and $\hat{E}_i^m$ represent model-based predicted ship's north and east positions at ith step future, respectively. n_T denotes the number of time steps of the prediction horizon. In this study, the Euler method is employed for the numerical integration with 1s time step; namely, $n_T = T$.

As shown in Fig. 4, this study expresses trajectories in the $x-y$ coordinate of which origin is located at the center of gravity of the ship when making a prediction. The positive directions of the x and y axes are the longitudinal and lateral directions of the ship. Thereby, future positions in the $x-y$ and $N-E$ coordinates are interconvertible as:

$$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} \cos\psi_0 & \sin\psi_0 \\ -\sin\psi_0 & \cos\psi_0 \end{pmatrix} \begin{pmatrix} N - N_0 \\ E - E_0 \end{pmatrix} \tag{11}$$

where N_0, E_0, and ψ_0 represent the north, east positions and heading when making a prediction. Hence, the model-based predicted trajectory $[\hat{N}_1^m, \ldots, \hat{N}_{n_T}^m, \hat{E}_1^m, \ldots, \hat{E}_{n_T}^m]$ in the $N-E$ coordinate is converted to $[\hat{x}_1^m, \ldots, \hat{x}_{n_T}^m, \hat{y}_1^m, \ldots, \hat{y}_{n_T}^m]$ in the $x-y$ coordinate to provide target vectors of the data-driven compensator.

3.3. Data-driven compensator

The data-driven compensator makes prediction of $[\Delta\hat{x}_1^m, \ldots, \Delta\hat{x}_{n_T}^m, \Delta\hat{y}_1^m, \ldots, \Delta\hat{y}_{n_T}^m]$ where $\Delta\hat{x}_i^m = x_i - \hat{x}_i^m$ and $\Delta\hat{y}_i^m = y_i - \hat{y}_i^m$, respectively, based on onboard sensor measurements. x_i and y_i are the true position at the i-step future. This target vector is given not in the model deployment but in the model training.

3.3.1. Input features

Input features of any ML models must be carefully selected. Otherwise, ML models would suffer from missing important information. In such settings, ML models fail to be efficiently trained. This study selects input features based on the theory of ship dynamics that future trajectories are determined by inertial and hydrodynamic parameters of the ship, the initial state of the ship, commands to thrusters, and environmental disturbances. As we develop data-driven compensators for the specific loading condition of the specific ship, the inertial and hydrodynamic parameters are assumed to be constant. Wave and ocean current data are not included in input features as they are mostly not measured in real time. We assume commands to thrusters, the true wind velocity, and the true wind direction are kept unchanged over the prediction horizon. These assumptions yield the formulation of the data-driven compensator f_N:

$$[\Delta\hat{x}_1^m, \dots, \Delta\hat{x}_{n_T}^m, \Delta\hat{y}_1^m, \dots, \Delta\hat{y}_{n_T}^m] = f_N(\boldsymbol{\nu_0}, n_0^p, \delta_0^p, n_0^s, \delta_0^s, u_{rw,0}, v_{rw,0}) \tag{12}$$

The suffix 0 represents the values when making a prediction. The input vector is z-score normalized with the statistic values in the training dataset. The same values are applied to the normalization in the validation and test datasets. In the experiments of this study, two azimuth thrusters are manipulated with the same commands. Thereby, (12) is reduced to:

$$[\Delta\hat{x}_1^m, \dots, \Delta\hat{x}_{n_T}^m, \Delta\hat{y}_1^m, \dots, \Delta\hat{y}_{n_T}^m] = f_N(\boldsymbol{\nu_0}, n_0, \delta_0, u_{rw,0}, v_{rw,0}) \tag{13}$$

where $n_0 = n_0^p = n_0^s$ and $\delta_0 = \delta_0^p = \delta_0^s$.

3.3.2. Model training

This study employs a MultiLayer Perceptron (MLP), a fully-connected feedforward neural network, which is one of the most classic architectures of neural networks. It consists of an input layer, hidden layer(s), and an output layer. tanh and linear functions are used for the hidden layer(s) and the output layer, respectively. Weights and biases are updated in the manner of the backpropagation by using Adam (Kingma and Ba, 2015) optimizer so that it minimizes the mean squared error between MLP's output and target vectors. During training, we separate some maneuvers from a training–validation dataset and keep them as a validation dataset. The validation loss is monitored to avoid overfitting the training dataset. If the validation loss does not improve over 200 epochs, the training is automatically terminated, and the best model is used for the prediction (early stopping). In this study, we build an MLP in the Pytorch (Paszke et al., 2019) framework in Python.

3.3.3. Hyperparameter tuning

Hyperparameters are parameters to be fixed in advance to determine ML model's architecture and training setting. A hyperparameter tuning is important to achieve a good performance of ML models. In this study, the number of hidden layers $\in [1, 3]$, the number of units in hidden layer(s) $\in [10, 500]$, the drop-out rate in the input layer $\in [0.0, 1.0]$, the drop-out rate in hidden layer(s) $\in [0.0, 1.0]$, and the learning rate of the optimizer $\in [10^{-5}, 10^{-1}]$ are optimized. The hyperparameter tuning is an optimization problem finding the best set of hyperparameters that performs the best in the validation dataset. In this study, such an optimum set is searched by using the Tree-structured Parzen Estimator (TPE) optimizer in the optuna (Akiba et al., 2019) framework. The TPE optimizer is one of the Bayesian optimization methods. It has been widely used for the hyperparameter tuning with a great performance and small computational time. Thereby, in the optuna, the TPE is selected as a default algorithm. In this section, details of the TPE algorithm are not revisited as it is not the focus of this study. For further information, original articles (Bergstra et al., 2013, 2011) for the TPE can be referred. The number of trials for the parameter search is 50 as further drastic improvement of the validation loss was not found with the larger number of trials than 50. The learning rate is searched in the log domain. After 50 trials of the hyperparameter search, a set of hyperparameters with the best performance in the validation dataset was selected as a set of optimum hyperparameters. Hyperparameter tuning was conducted independently for having different physics-based models and dataset.

3.3.4. Model deployment

In the model deployment, the data-driven compensator makes prediction $[\Delta\hat{x}_1^m, \dots, \Delta\hat{x}_{n_T}^m, \Delta\hat{y}_1^m, \dots, \Delta\hat{y}_{n_T}^m]$ based on input vectors provided by onboard sensors. By adding the model-based predicted position vector $[\hat{x}_1^m, \dots, \hat{x}_{n_T}^m, \hat{y}_1^m, \dots, \hat{y}_{n_T}^m]$, the cooperative prediction yields $[\hat{x}_1, \dots, \hat{x}_{n_T}, \hat{y}_1, \dots, \hat{y}_{n_T}]$ where $\hat{x}_i = \Delta\hat{x}_i^m + \hat{x}_i^m$ and $\hat{y}_i = \Delta\hat{y}_i^m + \hat{y}_i^m$. It is re-converted to the position vector in the $N - E$ coordinate by (11).

3.4. Evaluation metrics

The accuracy of the physics-based model A is evaluated with the Root Mean Squared Error (RMSE) of the geometrical similarity between true and predicted trajectories in the test dataset.

$$A = \frac{1}{S}\sum_{k=1}^{S}\sqrt{(N_k - \hat{N}_k^m)^2 + (E_k - \hat{E}_k^m)^2} \tag{14}$$

N_k and E_k represent the true north and east positions of the kth sample. $\hat{N}_k^m$, and $\hat{E}_k^m$ are the north and east positions of the kth sample predicted by the physics-based model. S is the number of data samples. N_D represents the number of maneuvers in the dataset used for the training. The larger the N_D is, the larger the dataset is. Hence, cooperative models are characterized by the combination (A, N_D) in this study. It should be noted that the ship dynamics is highly complex and nonlinear, thereby, a single metric A does not fully represent the characteristic of the physics-based model.

The errors made by the cooperative model H of (A, N_D) is evaluated with the RMSE of the geometrical similarity between the true and predicted trajectories made by the cooperative model:

$$H = \frac{1}{S}\sum_{k=1}^{S}\sqrt{(N_k - \hat{N}_k)^2 + (E_k - \hat{E}_k)^2} \tag{15}$$

where $\hat{N}_k$ and $\hat{E}_k$ are the north and east positions of the kth sample, predicted by the cooperative model. It should be noted that A is not used for selecting a good physics-based model for better cooperative performance since it is defined in the test dataset. A is only employed for presenting the relationship between H and (A, N_D) in the test dataset to develop a better understanding of the contribution of the physics-based model and data to the cooperative performance. If $N_D = 0$, we substitute $\hat{N}_k^m$ and $\hat{E}_k^m$ for $\hat{N}_k$ and $\hat{E}_k$ in graphs since the cooperative model without using any data is regarded as the physics-based model. If the pure data-driven model is used, $\hat{N}_k$ and $\hat{E}_k$ are calculated without the help of the physics-based model. In the application, large prediction errors become an issue since they could negatively affect our decision making ending up with fatal consequences. Thereby, we define the 90% of percentile of $\sqrt{(N_k - \hat{N}_k)^2 + (E_k - \hat{E}_k)^2}$ as H_{90} to examine the occurrence of large prediction errors.

4. Simulation experiment

Before performing full-scale experiments in Section 5, we examine the contribution of the physics-based model and data to the cooperative performance in the simulation environment. In the simulation environment, we can build physics-based models with different accuracy in a flexible manner since the ground-truth model is known. In addition, we can efficiently investigate the impact of data amount on the cooperative performance as we can generate virtual maneuvers as much as we need. Since trajectory predictions are used for the purpose of the early

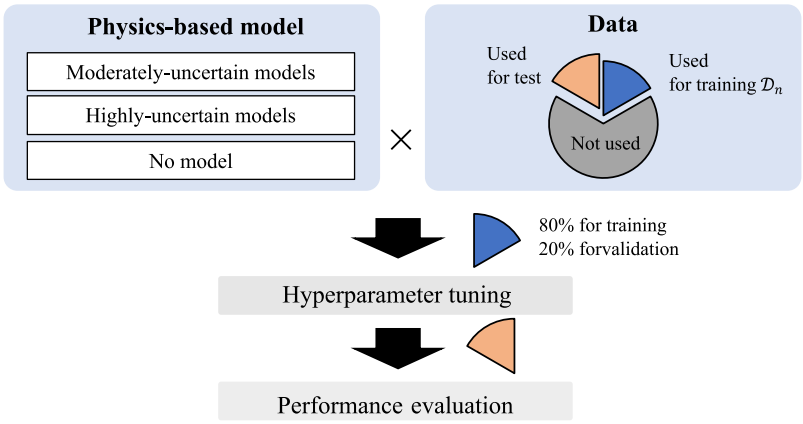


Fig. 5. Experimental setting.

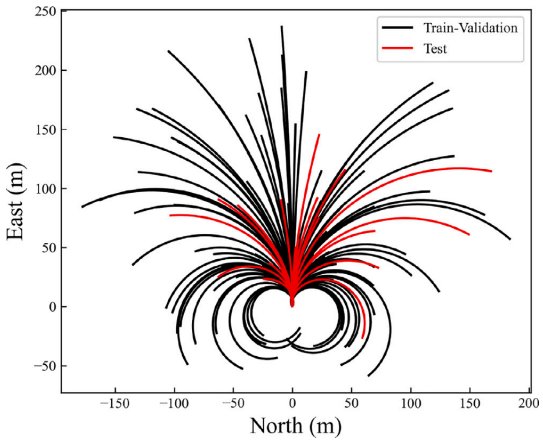


Fig. 6. Trajectories of maneuvers employed in the simulation experiment. Black trajectories show maneuvers grouped into the training–validation dataset and red trajectories show maneuvers grouped into the test dataset. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

warning of the collision risk, T, which is longer than the time that the ship can take evasive actions, is preferable for the evaluation of ship dynamic models. In the simulation experiment, the cooperative model was trained and evaluated for making $T = 30$s trajectory prediction.

4.1. Overview

The overview of simulation experiments in this study is illustrated in Fig. 5. 18 different physics-based models were used in the cooperative models. A dataset with 120 maneuvers was prepared in this study. They are explained in detail in the following subsections. 20 maneuvers in the dataset were randomly selected for the test dataset and kept untouched during training and validation of the cooperative models. By selecting N_D maneuvers from the remaining 100 maneuvers, we built the training and validation sub dataset $\mathcal{D}_{N_D}$ with the different number of maneuvers. Please note that $\mathcal{D}_{N_D=a} \subset \mathcal{D}_{N_D=b}$ if $a < b$. In this study, ten sub datasets from $\mathcal{D}_{N_D=10}$ to $\mathcal{D}_{N_D=100}$ were prepared. We trained the cooperative models with different physics-based models and sub datasets; thus examining the impact of the accuracy of the physics-based model and dataset on the cooperative performance. For the different combinations of the physics-based models and sub datasets, hyperparameter tuning was conducted independently. In $\mathcal{D}_{N_D}$, 80% of maneuvers were used for the training and the remaining 20% were used for the validation. The performance of the trained cooperative model was examined by using 20 maneuvers in the test dataset. The test dataset was always identical regardless of which sub dataset was used during training.

4.2. Dataset

A simulation dataset was generated by using a six Degrees of Freedom (DoF) seakeeping and maneuvering model of the R/V Gunnerus, which is a 28.9m-length Norwegian University of Science and Technology (NTNU)'s research ship. It is a high-fidelity ship dynamic model provided in the Open Simulation Platform project, which is a joint project with Kongsberg Maritime, DNV, SINTEF, and NTNU. It is composed of a hull model (Ross, 2008; Hassani et al., 2015) and thruster models running on the simulation platform Vico (Hatledal et al., 2021). Two azimuth thrusters were manipulated simultaneously. 120 unique turning maneuvers were generated by randomly selecting thruster revolution $n \in [50, 200]$ Revolution Per Minute (RPM), and thruster angle $\delta \in [-50, 50]°$. The ship's motion was disturbed by the constant wind and irregular waves in the simulation. The true wind direction $\beta_w \in [0, 360)°$, the true wind speed $V_w \in [0, 6]$ m/s, and the global wave direction $\in [0, 360)°$ are randomly chosen for each maneuver. The wave spectrum was JONSWAP spectrum (Hasselmann et al., 1973) with 1.0 m significant wave height and 5.0s significant wave period. The time step of the simulation environment was 0.05s. 50 s time series were saved in 1 Hz with 30 s future trajectory at each time step for each maneuver. Future trajectories were used only for training and evaluation purposes. Generated trajectories are shown in Fig. 6. A snapshot of a simulation experiment is shown in Fig. 2(a). Minimum, mean, and maximum input values of the data-driven compensator in datasets are shown in Table 1.

4.3. Physics-based models

In this experiment, cooperative models were trained with different physics-based models. They were developed by shifting parameters of the ground-truth model used in the simulation environment. This procedure introduced the model's uncertainty that we had in reality due to poorly identified parameters. We randomly produced 18 physics-based models with parameter uncertainty to examine the impact of the

Table 1
Minimum, mean, and maximum input values in the sub datasets for the training and validation and the test dataset in the simulation experiment.

		$\mathcal{D}_{10}$	$\mathcal{D}_{20}$	$\mathcal{D}_{30}$	$\mathcal{D}_{40}$	$\mathcal{D}_{50}$	$\mathcal{D}_{60}$	$\mathcal{D}_{70}$	$\mathcal{D}_{80}$	$\mathcal{D}_{90}$	$\mathcal{D}_{100}$	Test
u_0 (m/s)	min	0.3	0.3	0.1	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.1
	mean	2.2	2.3	2.2	2.2	2.2	2.2	2.3	2.4	2.4	2.4	1.9
	max	5.4	5.4	5.4	5.4	5.8	5.8	5.8	6.0	6.0	6.0	5.3
v_0 (m/s)	min	−0.9	−0.9	−0.9	−1.1	−1.1	−1.1	−1.1	−1.1	−1.1	−1.1	−0.8
	mean	0.0	0.0	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	max	1.0	1.0	1.0	1.0	1.1	1.1	1.1	1.1	1.1	1.1	0.8
r_0 (°/s)	min	−8.2	−8.2	−8.2	−8.2	−8.2	−8.2	−8.2	−8.2	−8.2	−8.2	−3.9
	mean	−0.5	−0.3	−0.4	−0.1	−0.1	−0.1	0.0	0.1	−0.1	0.0	0.2
	max	6.4	7.8	7.8	7.8	7.8	7.8	7.8	7.8	7.8	7.8	5.2
n_0 (RPM)	min	62.8	62.8	55.9	50.4	50.4	50.4	50.4	50.4	50.4	50.4	50.0
	mean	133.4	133.3	126.9	126.2	122.7	123.6	125.2	128.9	130.1	129.4	103.9
	max	191.7	192.4	192.4	192.4	199.6	199.6	199.6	199.6	199.6	199.6	189.1
δ_0 (°)	min	−37.3	−45.8	−45.8	−47.4	−47.4	−47.4	−47.4	−48.7	−48.7	−48.7	−38.7
	mean	1.7	0.8	1.9	−0.4	−0.7	−2.3	−2.6	−3.6	−2.1	−2.7	0.7
	max	48.3	48.3	48.3	48.3	48.3	48.3	48.3	48.3	49.4	49.4	41.4
$u_{rw,0}$ (m/s)	min	−4.0	−4.0	−4.0	−4.0	−4.2	−4.2	−4.4	−4.4	−4.4	−4.4	−3.0
	mean	2.9	3.1	3.1	2.7	2.5	2.5	2.6	2.7	2.8	2.8	2.5
	max	7.5	10.3	10.3	10.3	10.3	10.3	10.3	10.3	10.4	10.4	7.4
$v_{rw,0}$ (m/s)	min	−5.1	−5.4	−6.0	−6.0	−6.0	−6.0	−6.0	−6.0	−6.0	−6.0	−6.4
	mean	0.3	0.1	−0.1	−0.1	0.0	0.0	−0.2	−0.1	0.1	0.0	−0.4
	max	5.9	5.9	5.9	5.9	5.9	5.9	5.9	5.9	6.3	6.3	5.8

accuracy of the physics-based model on the cooperative performance. 18 models were grouped into moderately- and highly-uncertain models. It should be noted that none of them were identical to the ground-truth model used in the simulation. They produced prediction errors at different levels due to different reasons.

4.3.1. Moderately-uncertain models

We prepared ten physics-based models by randomly shifting parameters of the ground-truth model in $\boldsymbol{D}(\boldsymbol{\nu}_r)$. They were grouped into moderately-uncertain models in this paper. The ground-truth model has 32 hydrodynamic derivatives $\theta_1 - \theta_{32}$ in $\boldsymbol{D}(\boldsymbol{\nu}_r)$ (such as X^L_{uu}, see Ross et al. (2015) for details). A set of disturbed parameters $\theta'_{i,j}$ the ith hydrodynamic derivative of the jth moderately-uncertain model was introduced as:

$$\theta'_{i,j} = \Delta_{i,j}\theta_i \tag{16}$$

$0.4 < \Delta_{i,j} < 1.6$ was randomly selected for the ith hydrodynamic derivative of the jth moderately-uncertain model. Although they made prediction errors due to poorly identified parameters, predicted trajectories they made could represent the basic characteristics of the true dynamics of the targeting ship. The mass, inertia moment, added-mass coefficients, and thruster models were kept unchanged from the ground-truth model. This situation could occur if we copy and paste hydrodynamic parameters of similar ships, we have a physics-based model adjusted to the different operational conditions, and so on.

4.3.2. Highly-uncertain models

We prepared another eight physics-based models by randomly shifting the mass, inertia moment, added-mass coefficients, and propeller diameter of the thruster model up to 40%, in the same procedure as (16), in addition to the parameter shift introduced in the moderately-uncertain models. The trajectories they predicted had very different characteristics from the true trajectories since the basic parameters of the model were shifted. This situation could occur if we copy and paste parameters of very different ships or actuator models have large uncertainty.

4.3.3. No model

If no model was assigned to the physics-based model, a pure data-driven model was built in the experiment. It was trained in the same manner as the cooperative model, however, the target vector was not the residual vector $[\Delta\hat{x}^m_1, \ldots, \Delta\hat{x}^m_{n_T}, \Delta\hat{y}^m_1, \ldots, \Delta\hat{y}^m_{n_T}]$ but the future position vector $[\hat{x}_1, \ldots, \hat{x}_{n_T}, \hat{y}_1, \ldots, \hat{y}_{n_T}]$ without the help of any model-based guides.

4.4. Results

Snapshots in Fig. 7 show predictions with different physics-based models with different sub datasets at one of the example prediction time instance in the test dataset. Black dotted lines show the 30 s true trajectories, which are the same in the three subfigures. In Fig. 7(a), predicted trajectories made by pure data-driven models trained with $\mathcal{D}_{10}$, $\mathcal{D}_{30}$, and $\mathcal{D}_{100}$ are shown. Since they were not supported by any prior knowledge of ship dynamics, it is seen that they needed a large dataset to make predictions accurately. Models trained with $\mathcal{D}_{10}$ and $\mathcal{D}_{30}$ ended up making discontinuous trajectories with less similarity to the true trajectory. In Fig. 7(b), predicted trajectories made by cooperative models with one of the moderately uncertain physics-based models with $\mathcal{D}_{10}$, $\mathcal{D}_{30}$, and $\mathcal{D}_{100}$ are shown. Although physics-based models made prediction error in (b), it was rather small and captured the basic geometry of the true trajectory. It is seen that data-driven compensators compensated for such errors well only by using a small dataset $\mathcal{D}_{10}$. As the pure data-driven model with $\mathcal{D}_{10}$ failed at making an accurate prediction in (a), it implies that the physics-based model successfully supported the cooperative performance. In Fig. 7(c), predicted trajectories made by cooperative models with one of the highly-uncertain physics-based models with $\mathcal{D}_{10}$, $\mathcal{D}_{30}$, and $\mathcal{D}_{100}$ are shown. In (c), a trajectory predicted by the physics-based model notably diverges from the true trajectory. The poor performance of the physics-based model was induced by its parameters with higher uncertainty. We see it deteriorated the cooperative performance with $\mathcal{D}_{10}$ and $\mathcal{D}_{30}$ significantly while the performances in (b) were very good with the same datasets. Moreover, although the large prediction error was mitigated by having a large sub dataset $\mathcal{D}_{100}$, the cooperative performance with $\mathcal{D}_{100}$ in (b) outperforms that in (c).

An overview of results is illustrated in Fig. 8. As addressed in Section 3.4, this figure shows the relationship between the errors made by the cooperative model H as a height of bars and its (A, N_D) as a position of bars on the bottom plane, where A denotes the errors made by the physics-based model and N_D denotes the number of maneuvers in the training dataset. For instance, the height of the bar located at the small A and large N_D on the bottom plane shows the errors made by the cooperative model with a combination of such an accurate physics-based model and a large dataset for the training. Bars at $N_D = 0$ shows the original performance of the physics-based model without using any data for the training. We see that most cooperative models with highly-uncertain physics-based models made larger errors than cooperative

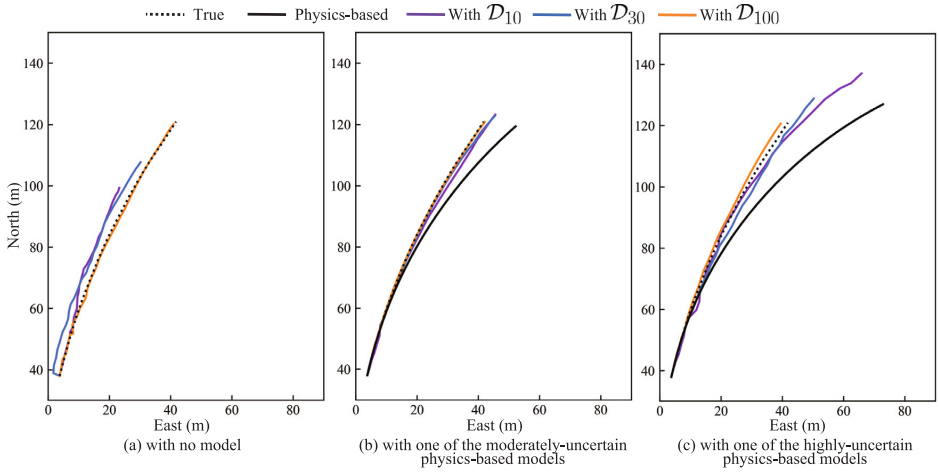


Fig. 7. Snapshots of trajectory predictions made by (a) the pure data-driven model, (b) the cooperative model with one of the moderately-uncertain physics-based models, and (c) the cooperative model with one of the highly uncertain physics-based models with sub dataset D_{10}, D_{30}, and D_{100}.

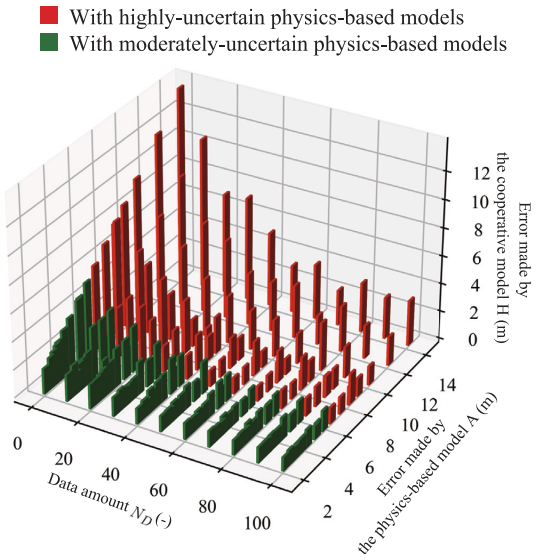


Fig. 8. The effect of the errors of the physics-based model A and the data amount N_D on the errors made by the cooperative model H in the test dataset.

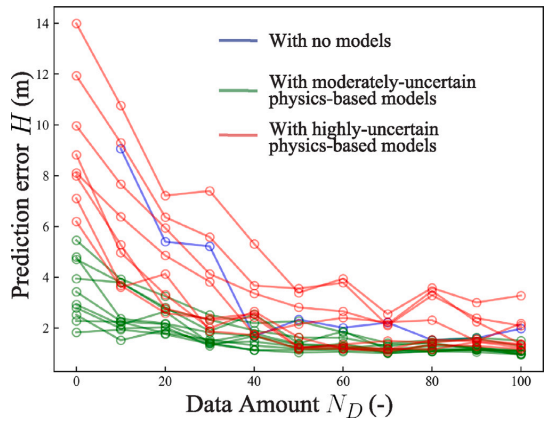


Fig. 9. A projected 2D graph of the effect of the accuracy of the physics-based model A and the data amount N_D on the errors made by the cooperative model H in the test dataset.

models with moderately-uncertain physics-based models with the same amount of data for training. In addition, a trend was seen that the higher cooperative performance was achieved with a larger dataset and a more accurate physics-based model. A good cooperative performance was achieved by either having an accurate physics-based model or having a large dataset; thereby, they are complementary to each other to some extent.

Fig. 9 is a projected graph of Fig. 8 on the $N_D - H$ plane for better visibility of absolute values of the cooperative performance. In Fig. 9, a trend is seen that the higher the accuracy of the physics-based model was, the higher the cooperative performance was, especially when the dataset was small. At the same time, cooperative models with a wide range of physics-based models, including some highly-uncertain models with relatively better performance, were found to outperform the pure data-driven model. It implies the possibility of the cooperative framework of building an accurate model with a compromised physics-based model and a small dataset. However, it does not mean any physics-based models are acceptable as a foundation of cooperative models. It is clearly seen that some cooperative models with highly-uncertain physics-based models ended up with poorer performance than the pure data-driven models. In such cases, physics-based models seem to not introduce prior knowledge of ship dynamics but introduce disturbances in the training. Thereby, the negative impact of having such physics-based models with the poor performance on the cooperative performance remained even if we had a large dataset. In particular, the performance of the cooperative model with the most inaccurate physics-based model fluctuated much depending on the sub dataset used for the training. It is seemingly caused by its high data dependency with a physics-based model introducing disturbance to the training.

Fig. 10 shows the relationship between N_D and H_{90} of models. Its trend is similar to that in Fig. 9. That means findings in Fig. 10 are

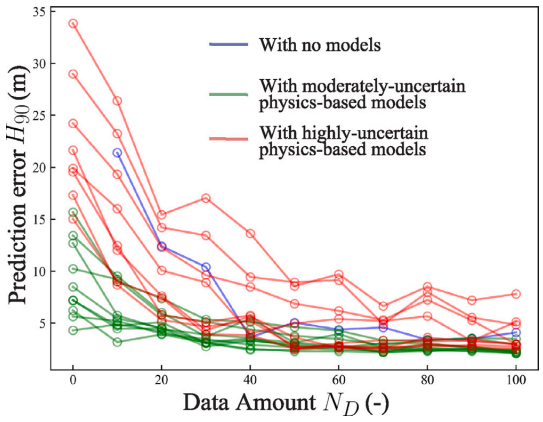


Fig. 10. A projected 2D graph of the effect of the accuracy of the physics-based model A and the data amount N_D on the 90% percentile cooperative performance H_{90} in the test dataset.

also applied to the strategy of how we reduce the occurrence of large prediction errors by using the physics-based model and data in this experiment.

5. Full-scale experiment

Hereinafter, we further explore the reasonable range of the physics-based model's accuracy on the cooperative performance in the real-life project by employing a small dataset of a full-scale experiment in the open sea. This full-scale experiment validates that we can build an accurate ship dynamic model in the practical project by combining a compromised physics-based model and a small dataset rather than relying on either of them. A snapshot of the experiment is shown in Fig. 2(b).

5.1. Overview

We made $T = 15$s trajectory predictions in the full-scale experiment since thruster commands changed drastically over the prediction horizon longer than $T = 15$s in a full-scale zigzag maneuvers. In addition, in the full-scale experiment, having too much uncertainty from unexpectable environmental disturbances in the longer prediction horizon makes a fair comparison between dynamic models challenging. Due to this limitation, errors over the prediction horizon from 15 s to 30 s have been widely used as a metrics of the accuracy of ship dynamic models in the full-scale experiment (see Skulstad et al. (2021a) and Wang et al. (2021)). The zigzag maneuver is one of the maneuvers that the International Towing Tank Conference (the ITTC) recommends as a full-scale maneuvering trials procedure. During the zigzag maneuver, ship's heading swings from side to side. The detailed definition of the zigzag maneuver can be referred to The International Towing Tank Conference (2002). In the full-scale experiment, we investigated the cooperative performances with different physics-based models and a small dataset to examine the framework building an accurate model with a compromised physics-based model and a small dataset.

5.2. Dataset

We conducted full-scale experiments in the open sea on November 21st, 2019 in Trondheim, Norway. The 33.9m-length R/V Gunnerus was employed. Under the mild weather condition, we conducted 10°/10°, 15°/15°, 20°/20°, 25°/25°, and 30°/30° zigzag maneuvers with high ($n \approx 145$RPM) and low ($n \approx 125$RPM) surge velocities. Each maneuver was saved in 1 Hz and cut into 85 s time series with 15 s future positions at each time step. The number of sampled maneuvers was 16. A 20°/20° zigzag maneuver with the high surge velocity was kept untouched for the test dataset. This maneuver was not included in the other maneuvers in the training–validation dataset. During the full-scale experiment, onboard sensors provided the following measurements:

- Positions: North and East positions in the NED coordinate in addition to the heading.
- Velocities: The surge, sway, and yaw velocities.
- Commands: Thruster revolution and angle of the port- and starboard-azimuth thrusters.
- Wind: The true wind direction and velocity.

During maneuvers, same commands were given to the two azimuth thrusters and the bow thruster was turned off. Except for the maneuver in the test dataset, 15 maneuvers were used for the training. The three-fold cross validation was conducted by using 15 maneuvers in the training and validation dataset. Minimum, mean, and maximum input values of the data-driven compensator in datasets are shown in Table 2.

5.3. Physics-based models

In the full-scale experiment, we employed two physics-based models; namely, accurate and inaccurate physics-based models to examine the impact of having different physics-based models on the cooperative performance with a real-life small dataset.

5.3.1. Accurate physics-based model

Before the full-scale experiment, the R/V Gunnerus was elongated from 28.9 m to 33.9 m. However, the corresponding ship dynamic model has not been fully developed. In this study, we employ a ship dynamic model of the 28.9 m R/V Gunnerus before the elongation as a physics-based model since it well captures the dynamic behavior of the elongated R/V Gunnerus as well. It is referred to as the accurate physics-based model in this experiment. It represents an optimistic assumption that an accurate physics-based model is available in the project.

5.3.2. Inaccurate physics-based model

In an inaccurate physics-based model, we shifted dominant damping coefficients in addition to removing higher-order damping coefficients in $\boldsymbol{D}(\boldsymbol{\nu}_r)$. The inaccurate physics-based model represents a pessimistic assumption that the available physics-based model performs poorly due to different reasons.

5.3.3. No model

If no physics-based model was given, as we did in the simulation experiment, a pure data-driven model was built without the help of the inaccurate physics-based model.

5.4. Results

In Fig. 11, snapshots of 15 s trajectory predictions of the maneuver in the test dataset at $t = 20$s in (a), $t = 40$s in (b), and $t = 60$s in (c) are shown. Time histories of the thruster revolution and angle of this maneuver are shown in Fig. 12. In Fig. 11(a), although the cooperative model with the accurate physics-based model made smaller errors compared to the other models, all models deviated from the true trajectory notably. In the full-scale experiment, as the dataset used for training was limited, it was seen that cooperative models did not always make a good prediction seemingly due to the lack of experience during training. On the other hand, in (b) and (c), cooperative models notably reduced prediction errors made by the corresponding physics-based models and significantly outperformed the pure data-driven model. In

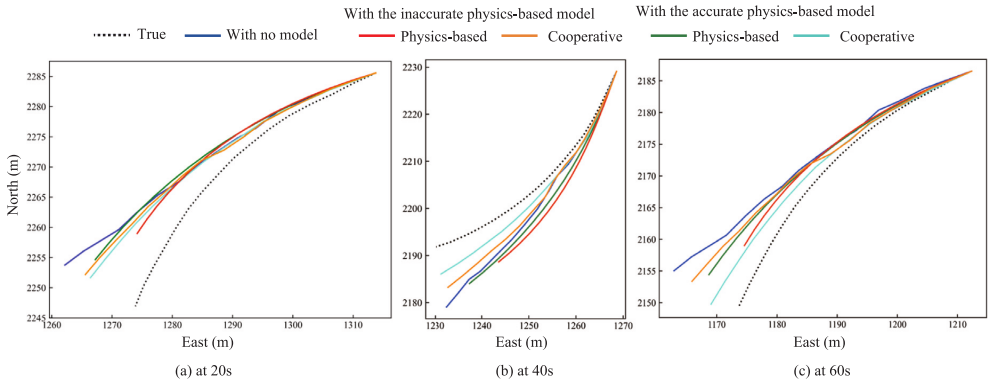


Fig. 11. Snapshots of trajectory predictions at (a) 20 s, (b) 40 s, and (c) 60 s of the maneuver in the test dataset.

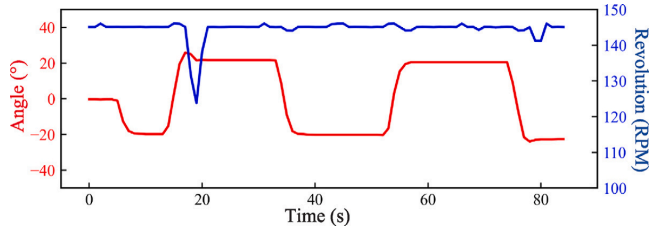


Fig. 12. A time series of thruster angle δ and revolution n in the example maneuvers in the test dataset shown in Fig. 11.

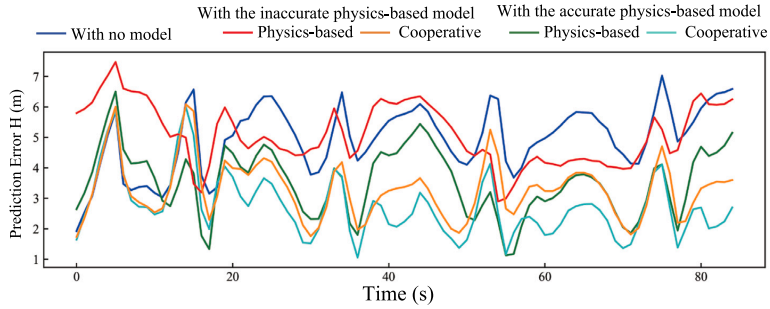


Fig. 13. Time series of the average prediction error H in the 15 s prediction horizon at each prediction instance of the maneuver in the test dataset.

(b), the cooperative model with the inaccurate physics-based model performed better than the accurate physics-based model. In (c), its performance was comparable to the accurate physics-based model. Hence, although the cooperative model with the inaccurate physics-based model did not outperform that with the accurate physics-based model, the contribution of having such a compromised physics-based model was clearly discerned. This finding corresponds to the results presented in the simulation experiment.

Time histories of the prediction error H for pure data-driven, pure physics-based, and cooperative models are shown in Fig. 13. The prediction performance fluctuated as time advanced due to having epistemic and aleatoric uncertainties. Thereby, it is seen that models did not make accurate predictions at some time steps, as shown in Fig. 11(a), however, the overall prediction performance of the cooperative model outperformed the pure data-driven and corresponding physics-based models. The performance of the cooperative model with the inaccurate physics-based model was mostly comparable with that of the accurate physics-based model.

Fig. 14 shows the average prediction error H in the 15 s prediction horizon at each prediction instance made by the pure data-driven, pure physics-based, and cooperative models. The prediction error becomes larger in the distant horizon as we have much uncertainty in the distant future. The pure data-driven model made larger prediction errors than the other models over the prediction horizon. It highlights the benefit of having the physics-based model in terms of prediction performance. Although the inaccurate physics-based model did not perform well in the experiment over the prediction horizon, its prediction error was well compensated for by using the small dataset. Its overall performance is comparable to that of the accurate physics-based model, which requires much time & effort to be developed. In addition, in Fig. 14, the discrepancy of the prediction error H made by the accurate and inaccurate physics-based model was found to be large. On the other hand, by using the small dataset, the discrepancy

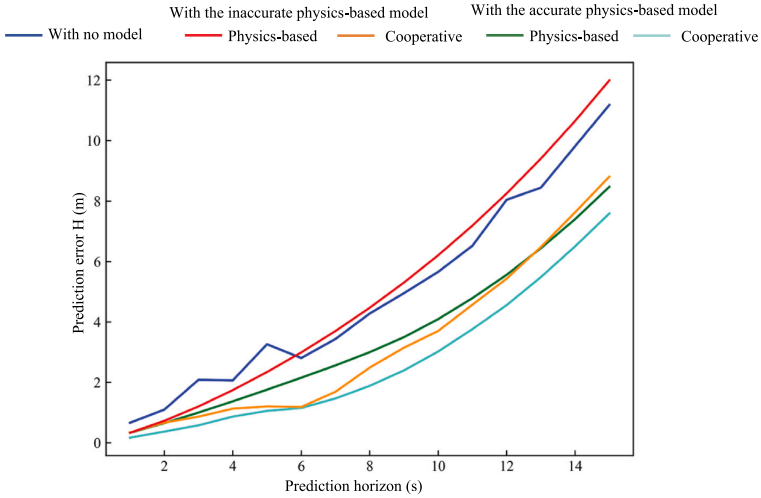


Fig. 14. The prediction error H over the prediction horizon made by pure data-driven, pure physics-based, and cooperative models.

Table 2
Maximum and Minimum input values in the training and validation dataset and the test dataset in the full-scale experiment.

		Training–validation	Test
u_0 (m/s)	min	2.4	3.5
	mean	4.1	3.8
	max	5.0	4.7
v_0 (m/s)	min	−0.6	−0.3
	mean	0.2	0.2
	max	0.8	0.7
r_0 (°/s)	min	−3.9	−3.9
	mean	−0.1	−0.1
	max	4.0	3.8
n_0 (RPM)	min	115.2	123.8
	mean	136.5	144.6
	max	145.1	146.1
δ_0 (°)	min	−29.7	−23.9
	mean	0.8	1.2
	max	29.8	25.9
$u_{rw,0}$ (m/s)	min	−3.3	−0.8
	mean	0.6	0.8
	max	7.5	2.8
$v_{rw,0}$ (m/s)	min	−5.8	−4.3
	mean	−0.8	−2.9
	max	6.0	−0.2

between corresponding cooperative performances becomes smaller. It implies the robustness of the cooperative model to the poor accuracy of the physics-based model.

6. Discussion

In this section, key findings in the simulation and full-scale experiments related to the open question "how do a physics-based model and data cooperate with each other in the cooperative model?" are summarized. In addition, this section clearly presents our suggestions for industrial applications based on the key findings. In the end, the limitations of this work and future works are discussed.

6.1. Key findings

In the simulation and full-scale experiments, **cooperative models were found to work better with a wide range of physics-based models than the pure data-driven models**. It indicates the cooperative model effectively introduced prior knowledge packaged in a wide range of physics-based models into the training in addition to acting as a stable foundation for making a prediction. The cooperative performance was notably improved by having larger datasets, thereby, the cooperative model was found to be robust to the poor accuracy of the physics-based model. On the other hand, the contribution of the physics-based model was critical when the dataset was small. If large datasets are available, the low accuracy of the physics-based model can be compensated to some extent.

In the cooperative models, **the most important perspective was to balance the physics-based model and data rather than solely relying on either of them.** In the experiments, in some cases, it was seen that the combination of a moderately-accurate physics-based model and a relatively-small dataset outperformed pure data-driven models with larger datasets or more highly-accurate physics-based models.

Although cooperative models were found to be robust to the poor accuracy of the physics-based model by being trained in a data-driven manner, **we have no such thing as a general ship model** that serves as a fundamental physics-based model of any ships. Simulation experiments in this study showed that a significantly inaccurate physics-based model may disturb the training of cooperative models instead of facilitating it with prior knowledge of ship dynamics. Its negative impact was found to remain even with a large dataset and it induced the instability of the performance due to them heavily relying on data with a disturbing foundation. Such models are not acceptable as a foundation and we should pay much attention to avoid using them.

Unlike the simulation experiment in Section 4, the full-scale experiment in Section 5 represents a real-life problem setting including the impact of real-world environmental disturbances and dataset. Thereby, the result in Section 5 validated the practical application of the framework of building an accurate model with a compromised physics-based model and a small dataset.

6.2. Suggestions for industrial applications

Findings in this study bring us to some suggestions for industrial applications. First, **we should devote more effort to developing methodologies that more easily and efficiently find highly-accurate physics-based models from the database.** It is expected for shipyards to accumulate results of numerical simulations, model/full-scale experiments, and hopefully identified parameters of the physics-based model of similar ships in their database. We could find a physics-based model of similar ships in the database in a more time- and cost-effective manner compared to the data collection; thereby, it would be a first priority in the model development. Having an accurate physics-based model makes the cooperative performance better, especially when a dataset is small. However, only a few studies (e.g., Mei et al. (2019)) have focused on the importance of this practice.

Second, **a key technology in the future would be how to build a simplified physics-based model easily**. Probably, we would not be able to find a physics-based model of similar ships from the database, especially when the project is carried out by small stakeholders without diverse experiences. In such settings, it is important to develop a physics-based model of which performance is in the acceptable range. With such a compromised model, we can improve the performance by collecting a limited amount of data as we showed in Section 5. It needs to be mentioned that we have criteria to be eligible for being a foundation model, although a wide range of physics-based models was found to be helpful. If the performance is too poor, it ruins the cooperative performance instead of helping. The co-simulation technology (e.g., Hatledal et al. (2021)) would be useful to build a simplified physics-based model only by assembling sub models. Such technology has not been fully applied to the maritime industry, however, it has great potential to provide a physics-based model with acceptable performance easily and readily.

6.3. Limitations

This study conducted experimental investigations by using different case studies in the simulation and full-scale experiments. Thereby, it is plausible to say that the findings in this study would provide basic insights for the practical applications of cooperative models. However, in future work, we must check how general our findings are in the theoretical, experimental, and practical manners involving more case studies.

This study employed a data-driven model without a physics-based model as a baseline. It does not represent a definite limitation of the data-driven approach itself. By using state-of-the-art ML architectures, pure data-driven models might perform better than we presented in this paper. However, in this field, the maritime industry does not apply such complex ML models without a physics foundation in their industrial applications due to the lack of the model's interpretability. In addition, such models would require great effort in their tuning and training. Thereby, good performance with a simple ML architecture could be seen as a practical benefit of having a physics-based model.

In the future, our research effort should be paid to developing methodologies how to balance our effort dedicated to having a physics-based model and collecting data. This study presented its basic understandings, however, it has not been revealed how to balance two methodologies in industrial practices and how it reduces the time & cost dedicated to the model development.

Cooperative models offer the development of the ship dynamic model in a timely and easy fashion for some applications including the situation awareness and onboard decision support. It can be also used as an initial-stage model of the project. However, it does not intend to substitute conventional ship dynamic models in all applications. For example, they would face a challenge in ensuring stability when they are fully implemented in the control system.

This study focused on challenges in building a ship dynamic model. Such a model is useful for a short-term prediction, estimating maneuverability, and building a simulator for training purposes. Dynamic models do not have information about the surrounding geography, traffic, and future environmental disturbances. Thereby, it should be noted that it is not suitable for making a long-term trajectory prediction such as 30 min.

7. Conclusion

In the era of ship automation in the future, precise ship dynamic models play an integral role in making the early warning of the future collision risks. In practice, it has been a great challenge how we develop such a model while minimizing the time & cost dedicated to the model development. A cooperative ship dynamic model, which employs a data-driven model for compensating for the position error made by the physics-based model, was presented in recent studies. It develops a ship dynamic model in a non-parametric manner exploiting data while having an interpretable and stable foundation of the physics-based model. Although it seems to be a promising direction to combine two approaches to overcome the time & cost challenges in industrial practices, it has been an open question "how much does the cooperative model benefit from physics knowledge and observation data?". This study conducted simulation and full-scale experiments to offer one solution through case studies and explore a safe zone of cooperative models in the physics-based model's accuracy and the data amount dimensions. In the simulation experiments, the performances of the cooperative models with different physics-based models and different datasets were examined. In addition, in the full-scale experiment, the impact of having different physics-based models on the cooperative performance with a real-life small dataset was investigated. Findings in the experiments showed that the balance of the accuracy of the physics-based model and the data amount is key to achieve a good performance of the cooperative model rather than relying on either of them. Although a wide range of physics-based models successfully facilitated the model identification, however, it disturbed the training if it was too inaccurate. In the full-scale experiment, a framework of building an accurate model with a compromised physics-based model and a small dataset was validated. Hence, for reducing the time & cost challenges in the cooperative framework, it would be pivotal to find an accurate physics-based model from database or build a simplified physics-based model with acceptable performance efficiently.

CRediT authorship contribution statement

Motoyasu Kanazawa: Conceptualization, Methodology, Investigation, Writing – original draft. **Tongtong Wang:** Conceptualization, Writing – review & editing. **Robert Skulstad:** Conceptualization, Supervision. **Guoyuan Li:** Writing – review & editing, Supervision. **Houxiang Zhang:** Writing – review & editing, Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgment

This work was supported by a grant from the Research Council of Norway, IKTPLUSS project No. 309323 "Remote Control Center for Autonomous Ship Support" in Norway.

References

Akiba, T., Sano, S., Yanase, T., Ohta, T., Koyama, M., 2019. Optuna: A next-generation hyperparameter optimization framework. In: KDD '19: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. Association for Computing Machinery, Anchorage, AK, USA, pp. 2623–2631. http://dx.doi.org/10.1145/3292500.3330701.

Bergstra, J., Bardenet, R., Bengio, Y., Balazs, K., 2011. Algorithms for hyper-parameter optimization. In: Advances in Neural Information Processing Systems (NIPS 2011). pp. 1–9.

Bergstra, J., Yamis, D., Cox, D., 2013. Making a science of model search: hyperparameter optimization in hundreds of dimensions for vision architectures. In: Proceedings of the 30th International Conference on Machine Learning. Vol. 28, Atlanta, Georgia, USA, http://dx.doi.org/10.1080/01459740.2015.1058375.

Chen, X., Ling, J., Wang, S., Yang, Y., Luo, L., Yan, Y., 2021. Ship detection from coastal surveillance videos via an ensemble canny-gaussian-morphology framework. J. Navig. 74, 1252–1266. http://dx.doi.org/10.1017/S0373463321000540.

Fonseca, A., Gaspar, H.M., 2021. Challenges when creating a cohesive digital twin ship: a data modelling perspective. 68, (2), pp. 70–83. http://dx.doi.org/10.1080/09377255.2020.1815140.

Hassani, V., Fathi, D., Ross, A., Sprenger, F., Selvik, Berg, T.E., 2015. Time domain simulation model for research vessel Gunnerus. In: Proceedings of the International Conference on Offshore Mechanics and Arctic Engineering - OMAE. Vol. 7, St.John's, Newfoundland, Canada, http://dx.doi.org/10.1115/OMAE201541786.

Hasselmann, K., Barnett, T.P., Bouws, E., Carlson, H., Cartwright, D.E., Eake, K., Euring, J.A., Gicnapp, A., Hasselmann, D.E., Kruseman, P., Meerburg, A., Mullen, P., Olbers, D.J., Richren, K., Sell, W., Walden, H., 1973. Measurements of wind-wave growth and swell decay during the joint North Sea wave project (JONSWAP). Deutsche Hydrogr. Zeitschrift 12 (Deutches Hydrographisches Institut).

Hatledal, L.I., Chu, Y., Styve, A., Zhang, H., 2021. Vico: An entity-component-system based co-simulation framework. Simul. Model. Pract. Theory 108 (September 2020), 102243. http://dx.doi.org/10.1016/j.simpat.2020.102243.

Kanazawa, M., Skulstad, R., Li, G., Hatledal, L.I., Zhang, H., 2021. A multiple-output hybrid ship trajectory predictor with consideration for future command assumption. IEEE Sens. J. 1–13. http://dx.doi.org/10.1109/JSEN.2021.3119069.

Kanazawa, M., Skulstad, R., Wang, T., Li, G., Hatledal, L.I., Zhang, H., 2022. A physics-data co-operative ship dynamic model for a docking operation. IEEE Sens. J. 22 (11), 11173–11183. http://dx.doi.org/10.1109/JSEN.2022.3171036.

Karniadakis, G.E., Kevrekidis, I.G., Lu, L., Perdikaris, P., Wang, S., Yang, L., 2021. Physics-informed machine learning. Nat. Rev. Phys. 3 (6), 422–440. http://dx.doi.org/10.1038/s42254-021-00314-5.

Karpatne, A., Atluri, G., Faghmous, J., Steinbach, M., Banerjee, A., Ganguly, A., Shekhar, S., Samatova, N., Kumar, V., 2016. Theory-guided data science: A new paradigm for scientific discovery from data. http://dx.doi.org/10.1109/TKDE.2017.2720168.

Kawan, B., Wang, H., Li, G., Chhantyal, K., 2017. Data-driven modeling of ship motion prediction based on support vector regression. In: Proceedings of the 58th Conference on Simulation and Modelling (SIMS 58) Reykjavik, Iceland, September 25th – 27th, 2017. Vol. 138, pp. 350–354. http://dx.doi.org/10.3384/ecp17138350.

Kingma, D.P., Ba, J.L., 2015. Adam: A method for stochastic optimization. In: 3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings. pp. 1–15.

Luo, W., Li, X., 2017. Measures to diminish the parameter drift in the modeling of ship manoeuvring using system identification. Appl. Ocean Res. 67, 9–20. http://dx.doi.org/10.1016/j.apor.2017.06.008.

Mei, B., Sun, L., Shi, G., 2019. White-black-box hybrid model identification based on RM-RF for ship maneuvering. IEEE Access 7, 57691–57705. http://dx.doi.org/10.1109/ACCESS.2019.2914120.

Norwegian Shipowners Association, 2019. Maritime outlook report. Technical Report, p. 56, URL https://maritimpolitikk.no/en/2021.

Panigrahi, S., Nanda, A., Swarnkar, T., 2021. A survey on transfer learning. Smart Innov. Syst. Technol. 194, 781–789. http://dx.doi.org/10.1007/978-981-15-5971-6_83.

Paszke, A., Gross, S., Bradbury, J., Lin, Z., Devito, Z., Massa, F., Steiner, B., Killeen, T., Yang, E., 2019. Pytorch : An imperative style, high-performance deep learning library. (NeurIPS).

Ross, A., 2008. Nonlinear manoeuvring models for ships: A lagrangian approach. In: Department of Engineering Cybernetics, Faculty of Information, Technology, Mathematics, and Electrical Engineering. (Ph.D. thesis). p. 181.

Ross, A., Hassani, V., Selvik, Fathi, D., 2015. Identification of nonlinear manoeuvring models for marine vessels using planar motion mechanism tests. In: Proceedings of the ASME 2015 34th International Conference on Ocean, Offshore and Arctic Engineering. St.John's, Newfoundland, Canada.

Schirmann, M.L., Collette, M.D., Gose, J.W., 2022. Data-driven models for vessel motion prediction and the benefits of physics-based information. Appl. Ocean Res. 120, http://dx.doi.org/10.1016/j.apor.2021.102916.

Skulstad, R., Li, G., Fossen, T.I., Vik, B., Zhang, H., 2021a. A hybrid approach to motion prediction for ship docking - integration of a neural network model into the ship dynamic model. IEEE Trans. Instrum. Meas. 70, http://dx.doi.org/10.1109/TIM.2020.3018568.

Skulstad, R., Li, G., Fossen, T.I., Wang, T., Zhang, H., 2021b. A co-operative hybrid model for ship motion prediction. Model. Identif. Control: Norwegian Res. Bull. 42 (1), 17–26. http://dx.doi.org/10.4173/mic.2021.1.2.

The International Towing Tank Conference, 2002. Ittc-recommended procedures: full scale measurements manoeuvrability full scale manoeuvring trials procedure. Technical Report.

van de Ven, P.W., Johansen, T.A., Sørensen, A.J., Flanagan, C., Toal, D., 2007. Neural network augmented identification of underwater vehicle models. Control Eng. Pract. 15 (6), 715–725. http://dx.doi.org/10.1016/j.conengprac.2005.11.004.

von Stosch, M., Oliveira, R., Peres, J., Feyo de Azevedo, S., 2014. Hybrid semi-parametric modeling in process systems engineering: Past, present and future. Comput. Chem. Eng. 60, 86–101. http://dx.doi.org/10.1016/j.compchemeng.2013.08.008.

Vonrueden, L., Mayer, S., Beckh, K., Georgiev, B., Giesselbach, S., Heese, R., Kirsch, B., Walczak, M., Pfrommer, J., Pick, A., Ramamurthy, R., Garcke, J., Bauckhage, C., Schuecker, J., 2021. Informed machine learning - a taxonomy and survey of integrating prior knowledge into learning systems. IEEE Trans. Knowl. Data Eng. 1–19. http://dx.doi.org/10.1109/TKDE.2021.3079836.

Wang, T., Li, G., Hatledal, L.I., Skulstad, R., Aesoy, V., Zhang, H., 2021. Incorporating approximate dynamics into data-driven calibrator: A representative model for ship maneuvering prediction. IEEE Trans. Ind. Inf. 1. http://dx.doi.org/10.1109/tii.2021.3088404.

Wang, Z., Zou, Z., Soares, C.G., 2019. Identification of ship manoeuvring motion based on nu-support vector machine. Ocean Eng. 183 (January), 270–281. http://dx.doi.org/10.1016/j.oceaneng.2019.04.085.

Xiao, Z., Fu, X., Zhang, L., Goh, R.S.M., 2020. Traffic Pattern Mining and Forecasting Technologies in Maritime Traffic Service Networks: A Comprehensive Survey. http://dx.doi.org/10.1109/TITS.2019.2908191.

Xiao, G., Wang, T., Chen, X., Zhou, L., 2022. Evaluation of ship pollutant emissions in the ports of los angeles and long beach. J. Mar. Sci. Eng. 10, 1206. http://dx.doi.org/10.3390/jmse10091206.

Xue, Y., Liu, Y., Ji, C., Xue, G., 2020. Hydrodynamic parameter identification for ship manoeuvring mathematical models using a Bayesian approach. Ocean Eng. 195 (November 2019), 106612. http://dx.doi.org/10.1016/j.oceaneng.2019.106612.

Zhang, X., Fu, X., Xiao, Z., Xu, H., Qin, Z., 2022. Vessel trajectory prediction in maritime transportation: current approaches and beyond. IEEE Trans. Intell. Transp. Syst. http://dx.doi.org/10.1109/TITS.2022.3192574.

E

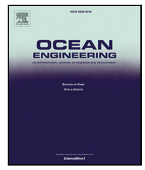
Paper V

Ocean Engineering 280 (2023) 114874

Contents lists available at ScienceDirect

Ocean Engineering

journal homepage: www.elsevier.com/locate/oceaneng



Bridging similar ships' dynamics for safeguarding the system identification of maneuvering models

Motoyasu Kanazawa [a,*], Tongtong Wang [a], Yasuo Ichinose [b], Robert Skulstad [a], Guoyuan Li [a], Houxiang Zhang [a]

[a] The Department of Ocean Operations and Civil Engineering, Norwegian University of Science and Technology, 6009 Ålesund, Norway
[b] The National Maritime Research Institute, 181-0004 Tokyo, Japan

ARTICLE INFO

Keywords:
Knowledge transfer
Maneuvering model
Parameter identification

ABSTRACT

System identification (SI) approach identifies ship maneuvering models using data from free-running maneuvers. As it is impossible to comprehensively validate model performance, a reliable model must be built within a consistent range with similar ships' models. However, in the SI approach, models are greatly influenced by the dataset design and feature selection, leading to the instability of model identification and the failure to ensure such consistency. To address this issue, our new idea introduces similar ship's model into model identification as knowledge foundation. First, we select a similar ship with three-step procedure, then build a new model by refining model parameters for a similar ship. Such a refinement is conducted with additional l2 regularization term in ridge regression, which balances knowledge and data-driven refinement. Designing the safe range of such refinement with hyperparameters, this study helps designers find a robust-and-accurate model within designed safe zones. In simulation experiments, we built knowledge connection between 161 m and 175 m container ships. By using the well-validated model of the former, a robust-and-accurate model for the latter was easily built by using a limited dataset, resulting in excellent model performance. This study makes the SI approach more promising by incorporating knowledge connections between similar ships.

1. Introduction

Marine accidents lead to human, economical, and environmental damages. To avoid such losses, the maritime industry has been giving marine officers adequate training and performing sufficient validation of control algorithms. Having critical scenarios is key for such training and validation, however, we hardly conduct real-world experiments due to safety and cost reasons. Thereby, simulators have provided a cyber world where we can experience an illusion of maneuvering. In the field of the aviation (Myers et al., 2018) and automobile (MartínelosReyes et al., 2019), in simulators, novice officers conduct basic training, and validation works of systems are performed. The maritime industry is no exception in this trend (Perez and Fossen, 2009).

For ensuring the quality of virtual maneuvering experiences, a fundamental element of marine simulators is ship maneuvering models. They describe hydrodynamic forces and moments acting on a ship in the form of polynomials with states of ship motion (velocities and accelerations) and control inputs. A complex interaction between a hull, propeller, and rudder is represented in coefficients of the polynomials. Mostly, maneuvering models have a large number of coefficients, which act as model parameters surrogating such complexity. Model parameters are dependent on multiple factors that include, but not limited to, hull shape, water depth, and ship draft. It is hard work for designers to identify parameters for a robust-and-accurate model for every ship for every condition, one by one, by conducting experiments. Two approaches are known for experimentally identifying the parameters; namely, Captive Model Test (CMT) and System Identification (SI) approaches (The International Towing Tank Conference, 2014). The CMT approach groups one Parameter Identification (PI) problem into several sub-PI problems composed of corresponding sub regression models and sub experiments. Although it has been regarded as the most reliable data source, it requires much expertise, effort, time, and special facilities. Thereby, for delivering models for many ships on simulators promptly, the SI approach gains attention recently. It identifies a large number of parameters all at once only with using a few trajectories of free-running maneuvers such as turning-circle and zigzag maneuvers without fixing a ship to a platform. Unlike the CMT approach, it delivers maneuvering models readily and at low cost. In addition, it is helpful to avoid scale-effect, which is a difference in model performance due to differences between model- and full-scale phenomena, by employing full-scale data in the SI approach.

* Corresponding author.
E-mail address: motoyasu.kanazawa@ntnu.no (M. Kanazawa).

https://doi.org/10.1016/j.oceaneng.2023.114874
Received 17 January 2023; Received in revised form 9 May 2023; Accepted 17 May 2023
Available online 29 May 2023

However, there exists a practical challenge that such an all-at-once PI of the SI approach leads to parameters having deceptive values due to multicollinearity between parameters. This challenge is known as parameter drift (Luo and Li, 2017). Once it happens, identified models lack robustness and accuracy. To avoid parameter drift, in the SI approach, designers have devoted significant effort in (1) data design: such as the optimization of excitation signals (Wang et al., 2020) and smoothing of experiment data (Wang et al., 2021b), (2) model design: optimizing a maneuvering model based on domain knowledge (Luo and Li, 2017) and feature-selection technique (Luo and Zou, 2009), and (3) PI algorithm design: selection and optimization of machine-learning algorithms for PI (Wang et al., 2021b). Unlike the CMT approach that highly restricts impacts of (1)–(3) within sub regression models, poor conduct of (1)–(3) in the SI approach could have catastrophic effects on model performance and robustness. In this field, models cannot be validated in a way that checks their performance all over the domain due to the prohibitive cost of data collection. Therefore, experts need to play active roles in certifying models based on whether they have sufficient confidence in model's validity (Sargent, 2010). A source of such confidence comes from the fact that models are built within a safe zone: the range where experts believe they are reasonable in light of the consistency between models for similar ships. Introducing the safe zone brings a stability to the PI while reducing risk that poor conduct of (1)–(3) easily leads to a catastrophic failure of modeling. The SI approach has been overlooking this perspective.

To offer one solution to this issue, this study proposes a new framework called **model refinement** in the SI approach by bridging two similar ships. When building a maneuvering model for a new ship with its free-running data, a model built in the CMT approach for a similar ship may be available. In this study, the former ship is referred to as a target ship being a target of the PI. The latter ship is referred to as a source ship acting as a knowledge safeguard. In such a case, designers would need to prioritize the model for the similar ship into building the new model for the new ship for having consistency between models for two similar ships. Otherwise, parameters are searched globally highly depending on the collected dataset. In addition, it is hard for experts to have sufficient confidence in model's validity if a new model is detached from connections between models. To prioritize such a connection in modeling and bridge dynamics of similar ships, this study employs parameters of the similar ship as a l_2 constraint in the ridge regression algorithm when identifying parameters for the new ship with free-running data of the new ship. It aims at refusing to update parameters into the range where designers do not justify such an update. Parameters of the similar ship are introduced as a safeguard in the data-driven calibration for controlling the degree of contribution of data infused into such a knowledge foundation. It helps designers find a robust-and-accurate model within the safe zone. The proposed framework gives models of similar ships the value as a knowledge asset bringing such benefits into building a new model for a new ship.

To validate the present framework, we conducted a case study on the widely-used Marine Systems Simulator (MSS) developed by Perez and Fossen (2009). We took an Abkowitz model (M.A., 1964) of a Mariner class ship, which is a 161m-length container ship, as a source ship. The Abkowitz model for the source ship has been developed with CMTs and widely-used. A target ship was a 175m-length SR108 container ship. Contributions of the present work are summarized as follows:

- Proposing a new framework that introduces parameters of a maneuvering model for a similar ship into building a new model for a new ship with free-running data. By using such a knowledge foundation, designers introduce a knowledge safeguard in the data-driven calibration for controlling the degree of contribution of data infused into such a knowledge foundation.
- Helping them easily find a robust-and-accurate model within the range defined by such a knowledge safeguard. It remedies effort dedicated to carefully designing experiments, models, and identification algorithms.
- Verifying the proposed approach in the simulation case study, which conducted a knowledge transfer between maneuvers of similar ships.

This paper unfolds as follows. In Section 2, previous works related to the PI and knowledge transfer of ship maneuvering models will be revisited. Section 3 illustrates the proposed framework. The case study will be presented in Section 4. Conclusions are given in Section 5.

2. Related works

In this section, previous works related to the PI with free-running data are revisited to clarify the novelty of the proposed framework bridging two ships' dynamics. As mentioned, the biggest challenge of the SI approach is parameter drift. This section briefly describes traditional approaches for the data design, model design, and PI algorithm design to alleviate it. In addition, this section also recapitulates recent works making a synergy of knowledge and data for building ship maneuvering models.

2.1. Traditional approaches for avoiding parameter drift

2.1.1. Data design

Since the SI approach highly relies on a collected dataset, it is of great importance to carefully design the dataset. Wang et al. (2020) optimized excitation signals during experiments such that collected datasets have rich information about ship dynamics. Ljungberg et al. (2022) discussed how to design experiments for identification of marine models. It should be noted that the experiment design is restricted by physical (e.g., width and length of experiment tanks) and time (e.g., experiment schedule) constraints in practice. Besides experiment designs, it is popular to denoise signals in the collected dataset by applying filtering techniques such as the Savitzky–Golay filter (Wang et al., 2021b) or the singular value decomposition (Xu et al., 2020).

2.1.2. Model design

To enhance the accuracy and robustness of maneuvering models, it is key to have only informative features for avoiding parameter drift. It has been conducted by removing redundant features by feature-selection techniques (Luo and Zou, 2009) and introducing modifications into models based on designers' knowledge (Luo and Li, 2017; Mucha and El Moctar, 2015).

2.1.3. PI algorithm design

By employing identification algorithms that are robust to noise, it is known that parameter drift can be alleviated to some extent. In particular, using Support Vector Machine (SVM) has been a popular approach thanks to its robustness against measurement noise (Wang et al., 2021b; Zhu et al., 2019). Researchers have also paid attention to properly tuning hyperparameters of identification algorithms (Luo et al., 2016).

2.2. Knowledge-data cooperative approach

Recently, it becomes popular to make a synergy of knowledge and observation data for modeling ship dynamics. A Bayesian framework updates a prior distribution by using data. The Kalman filter (KF) has been a popular approach based on the Bayes' theorem balancing knowledge represented by a dynamic model and data coming from measurements. Alexandersson et al. (2022) firstly denoised signals with a preliminary maneuvering model by the KF, and then employed the denoised signals for identifying a more sophisticated maneuvering model. By iterating this procedure, they built up a maneuvering model in a knowledge-data cooperative manner. For autonomous underwater vehicles, Cardenas and De Barros (2020) put extensive effort in developing analytical estimations for hydrodynamic derivatives. They were further

updated online by having the upcoming data by the KF. An objective of their approaches was to denoise signals and regularize the PI. In their study, newly-identified parameters could be drifted away from their prior estimations without introducing feedback from experts. However, given that data collection is never conducted comprehensively in this field, it seems not a wise step to overlook the role of knowledge for experts as a source of modeling confidence. For the CMT approach, the ITTC's guideline (The International Towing Tank Conference, 2014) suggests that designers should adjust model's parameters inside the range compatible with their uncertainty if the model does not perform satisfactorily during their validation. It is common to adjust parameters related to propellers and rudders. It aims at refusing manually updating parameters into the range where knowledge does not justify such an update. When bridging two ships' dynamics, we need to introduce this idea of safeguarding into the SI approach by using parameters for similar ships as a safeguard and controlling the degree of contribution of data being infused into such a knowledge foundation.

Among knowledge-data cooperative approaches, bridging dynamics of two ships have gained much attention. Designers invest a large amount of cost, time, and effort to develop a maneuvering model for one ship. Thereby, they would want to make use of such existing models for building a new model for a new ship. If models for similar ships have no relations, it would be hard for experts to have sufficient confidence in it. A connection between models for similar ships and conditions would be helpful for designers to have sufficient confidence in their validity. Cooperative models composed of a reliable mathematical model and a black-box model have acted as a harbinger of bridging two ships' dynamics. Mei et al. (2019) employed a random forest to calibrate the acceleration estimation made by a dynamic model of a similar ship. Wang et al. (2021a) bridged estimations of future positions of two ships by using a maneuvering model of the one ship and observation data of the other ship with a neural network. Cooperative models have achieved a great performance, however, their interpretability is still limited due to black-box models being involved as a part of them. This study aims at refining parameters of maneuvering models using data, thereby, it aligns with a traditional procedure of modeling without having black-box models in the model.

3. Knowledge transfer between similar ships

A schematic overview of the proposed framework is shown in Fig. 2. In the first subsection, we start with a general introduction of the Abkowitz model (M.A., 1964), which is one of the most widely-used maneuvering models in this field. Subsequently, the second subsection explains how we identify its parameters. Then, an integral part of the proposed framework: the model refinement will be explained. Here specifically the Abkowitz model and ridge regression are employed, however, this framework can be used for any types of maneuvering models and regularizations.

3.1. Ship maneuvering model

Assuming the propeller revolution is constant, the maneuvering theory of ship dynamics in 3DoF formulates the maneuvering motion in the surge, sway, and yaw directions as (Luo and Zou, 2009):

$$
\begin{aligned}
&(m - X_{\dot{u}})\dot{u} = f_1(u, v, r, \delta)\\
&(m - Y_{\dot{v}})\dot{v} + (mx_G - Y_{\dot{r}})\dot{r} = f_2(u, v, r, \delta)\\
&(mx_G - N_{\dot{v}})\dot{v} + (I_z - N_{\dot{r}})\dot{r} = f_3(u, v, r, \delta)
\end{aligned} \tag{1}
$$

where m is the ship mass, x_G is the longitudinal position of the ship's center of gravity, and I_z is the inertia moment about the vertical axis. u, v, and r denote the surge, sway, and yaw velocities in the body-fixed coordinate. δ represents the rudder angle. $X_{\dot{u}}$, $Y_{\dot{v}}$, $Y_{\dot{r}}$, $N_{\dot{v}}$, and $N_{\dot{r}}$ are the added-mass coefficients. f_1, f_2, and f_3 are functions of forces and moment acting on the ship. The small perturbations of the velocities and rudder angle are expressed as $\Delta u = u - U_0$, $\Delta v = v - v_0$, $\Delta r = r - r_0$, and $\Delta\delta = \delta - \delta_0$ where the subscript 0 represents variables at the initial instant. U_0 is the initial surge speed. In this study, the straightforward motion is given as an initial state; namely, $v_0 = 0$, $r_0 = 0$, and $\delta_0 = 0$. Variables in Eq. (1) are converted into nondimensional variables as the prime system (Fossen, 2011):

$$
\begin{aligned}
&\Delta u' = \frac{\Delta u}{U}, \Delta v' = \frac{\Delta v}{U}, \Delta r' = \frac{L\Delta r}{U}, \Delta\delta' = \Delta\delta\\
&\Delta\dot{u}' = \frac{L\Delta\dot{u}}{U^2}, \Delta\dot{v}' = \frac{L\Delta\dot{v}}{U^2}, \Delta\dot{r}' = \frac{L^2\Delta\dot{r}}{U^2}
\end{aligned} \tag{2}
$$

where $U = \sqrt{(U_0 + \Delta u)^2 + \Delta v^2}$ is the resultant speed and L is the ship length. Then, the Abkowitz model is introduced as:

$$
\begin{bmatrix} m' - X'_{\dot{u}} & 0 & 0 \\ 0 & m' - Y'_{\dot{v}} & m'x_G{}' - Y'_{\dot{r}} \\ 0 & m'x_G{}' - N'_{\dot{v}} & I'_z - N'_{\dot{r}} \end{bmatrix} \begin{bmatrix} \Delta\dot{u}' \\ \Delta\dot{v}' \\ \Delta\dot{r}' \end{bmatrix} = \begin{bmatrix} \Delta f'_1 \\ \Delta f'_2 \\ \Delta f'_3 \end{bmatrix} \tag{3}
$$

where the superscript $'$ denotes the nondimensional variables. The Abkowitz model gives functions of nondimensional forces and moment $\Delta f'_1$, $\Delta f'_2$, and $\Delta f'_3$ based on the form of the 3rd-order Taylor series expansion under the simplification employed in Fossen (2011):

$$
\begin{aligned}
\Delta f'_1 =& X'_u\Delta u' + X'_{uu}\Delta u'^2 + X'_{uuu}\Delta u'^3 + X'_{vv}\Delta v'^2 + X'_{rr}\Delta r'^2\\
&+ X'_{\delta\delta}\Delta\delta'^2 + X'_{\delta\delta u}\Delta\delta'^2\Delta u' + X'_{vr}\Delta v'\Delta r'\\
&+ X'_{v\delta}\Delta v'\Delta\delta' + X'_{v\delta u}\Delta v'\Delta\delta'\Delta u' + X'_{uvv}\Delta u'\Delta v'^2\\
&+ X'_{urr}\Delta u'\Delta r'^2 + X'_{uvr}\Delta u'\Delta v'\Delta r' + X'_{r\delta}\Delta r'\Delta\delta'\\
&+ X'_{ur\delta}\Delta u'\Delta r'\Delta\delta' + X'_0
\end{aligned} \tag{4}
$$

$$
\begin{aligned}
\Delta f'_2 =& Y'_{0u}\Delta u' + Y'_{0uu}\Delta u'^2 + Y'_v\Delta v' + Y'_r\Delta r' + Y'_\delta\Delta\delta'\\
&+ Y'_{vvv}\Delta v'^3 + Y'_{\delta\delta\delta}\Delta\delta'^3 + Y'_{vvr}\Delta v'^2\Delta r'\\
&+ Y'_{vv\delta}\Delta v'^2\Delta\delta' + Y'_{v\delta\delta}\Delta v'\Delta\delta'^2 + Y'_{\delta u}\Delta\delta'\Delta u'\\
&+ Y'_{vu}\Delta v'u' + Y'_{ru}\Delta r'\Delta u' + Y'_{\delta uu}\Delta\delta'\Delta u'^2 + Y'_{rrr}\Delta r'^3\\
&+ Y'_{vrr}\Delta v'\Delta r'^2 + Y'_{vuu}\Delta v'\Delta u'^2 + Y'_{ruu}\Delta r'\Delta u'^2\\
&+ Y'_{r\delta\delta}\Delta r'^2\Delta\delta' + Y'_{rr\delta}\Delta r'^2\Delta\delta' + Y'_{rv\delta}\Delta r'\Delta v'\Delta\delta' + Y'_0
\end{aligned} \tag{5}
$$

$$
\begin{aligned}
\Delta f'_3 =& N'_{0u}\Delta u' + N'_{0uu}\Delta u'^2 + N'_v\Delta v' + N'_r\Delta r' + N'_\delta\Delta\delta'\\
&+ N'_{vvv}\Delta v'^3 + N'_{\delta\delta\delta}\Delta\delta'^3 + N'_{vvr}\Delta v'^2\Delta r'\\
&+ N'_{vv\delta}\Delta v'^2\Delta\delta' + N'_{v\delta\delta}\Delta v'\Delta\delta'^2 + N'_{\delta u}\Delta\delta'\Delta u'\\
&+ N'_{vu}\Delta v'u' + N'_{ru}\Delta r'\Delta u' + N'_{\delta uu}\Delta\delta'\Delta u'^2 + N'_{rrr}\Delta r'^3\\
&+ N'_{vrr}\Delta v'\Delta r'^2 + N'_{vuu}\Delta v'\Delta u'^2 + N'_{ruu}\Delta r'\Delta u'^2\\
&+ N'_{r\delta\delta}\Delta r'^2\Delta\delta' + N'_{rr\delta}\Delta r'^2\Delta\delta' + N'_{rv\delta}\Delta r'\Delta v'\Delta\delta' + N'_0
\end{aligned} \tag{6}
$$

Five zero-frequency added-mass derivatives $X'_{\dot{u}}$, $Y'_{\dot{v}}$, $Y'_{\dot{r}}$, $N'_{\dot{v}}$, and $N'_{\dot{r}}$ are usually calculated precisely by using semi-empirical formulas or a strip-theory calculation, thereby, they are always estimated before the PI of the other derivatives (Luo and Zou, 2009). 16, 22, and 22 hydrodynamic derivatives in $\{X'_{(\cdot)}, Y'_{(\cdot)}, N'_{(\cdot)}\}$ are identified.

3.2. Regression model

The linear regression models in the surge, sway, and yaw directions are formulated in the discrete forms with the interval of the sampling time h as:

$$
\Delta u(k+1) - \Delta u(k) = AX \tag{7}
$$

$$
\Delta v(k+1) - \Delta v(k) = BY \tag{8}
$$

$$
\Delta r(k+1) - \Delta r(k) = CN \tag{9}
$$

(k) and $(k+1)$ denote velocities at the certain time step and those at the one-step ahead. $A = [a_1, \ldots, a_{16}]_{1\times16}$, $B = [b_1, \ldots, b_{22}]_{1\times22}$, and $C = [c_1, \ldots, c_{22}]_{1\times22}$ are the parameter vectors to be identified. The

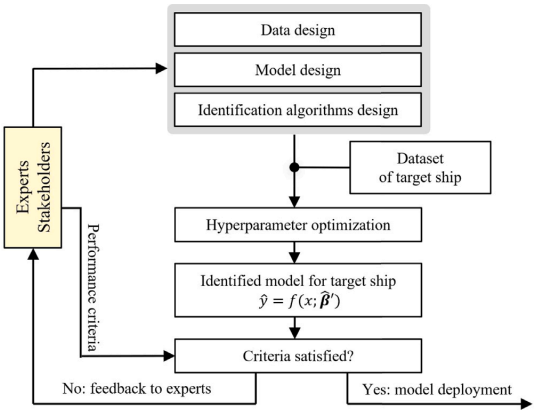


Fig. 1. An overview of a traditional data-based framework.

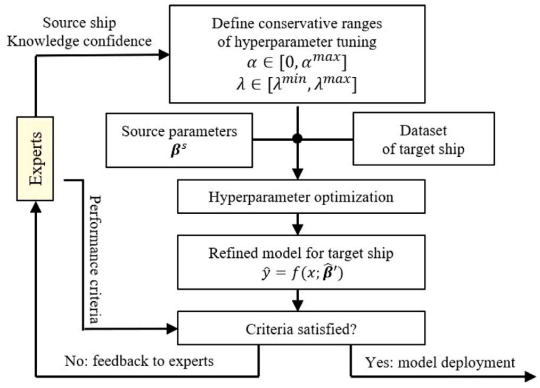


Fig. 2. An overview of the proposed framework.

input vectors of the linear regression models X, Y, and N are given as:

$$\begin{aligned} X = [&\Delta u(k)U(k), \Delta u^2(k), \Delta u^3(k)/U(k), \Delta v^2(k),\\ &\Delta r^2(k)L^2, \Delta\delta^2(k)U(k)^2, \Delta\delta^2(k)\Delta u(k)U(k),\\ &\Delta v(k)\Delta r(k)L, \Delta v(k)\Delta\delta(k)U(k), \Delta v(k)\Delta\delta(k)\Delta u(k),\\ &\Delta u(k)\Delta v(k)^2/U(k), \Delta r^2(k)L^2/U(k),\\ &\Delta u(k)\Delta v(k)\Delta r(k)L/U(k), \Delta r(k)\Delta\delta(k)LU(k),\\ &\Delta u(k)\Delta r(k)\Delta\delta(k)L, U^2(k)]^T_{1\times 16} \end{aligned} \tag{10}$$

$$\begin{aligned} Y = N = [&U^2(k), \Delta u(k)U(k), \Delta u^2(k), \Delta v(k)U(k), \Delta r(k)U(k)L,\\ &\Delta\delta(k)U^2(k), \Delta v^3(k)/U(k), \Delta\delta^3(k)U^2(k),\\ &\Delta v^2(k)\Delta r(k)L/U(k), \Delta v^2(k)\Delta\delta(k), \Delta v(k)\Delta\delta^2(k)U(k),\\ &\Delta\delta(k)\Delta u(k)U(k), \Delta v(k)\Delta u(k), \Delta r(k)\Delta u(k)L,\\ &\Delta\delta(k)\Delta u^2(k), \Delta r^3(k)L^3/U(k), \Delta v(k)\Delta r^2(k)L^2/U(k),\\ &\Delta v(k)\Delta u^2(k)/U(k), \Delta r(k)\Delta u^2(k)L/U(k),\\ &\Delta r(k)\Delta\delta^2(k)LU(k), \Delta r^2(k)\Delta\delta(k)L^2,\\ &\Delta r(k)\Delta v(k)\Delta\delta(k)L]^T_{1\times 22} \end{aligned} \tag{11}$$

In the PI, the left sides of (7)–(9) and input vectors are given. The PI algorithm finds the optimum A, B, and C so that they satisfy (7)–(9) well. Once they are identified, 16, 22, and 22 hydrodynamic derivatives for three directions $\{X'_{(\cdot)}, Y'_{(\cdot)}, N'_{(\cdot)}\}$ are further derived as:

$$X'_{(\cdot)} = \frac{L(m' - X'_{\dot{u}})}{h} A \tag{12}$$

$$\begin{bmatrix} Y'_{(\cdot)} \\ N'_{(\cdot)} \end{bmatrix} = \begin{bmatrix} \frac{(I'_z - N'_{\dot{r}})h}{SL} & -\frac{(m'x'_G - Y'_{\dot{r}})h}{SL} \\ -\frac{(m'x'_G - N'_{\dot{v}})h}{SL^2} & \frac{(m' - Y'_{\dot{v}})h}{SL^2} \end{bmatrix}^{-1} \begin{bmatrix} B \\ C \end{bmatrix} \tag{13}$$

where $S = (m' - Y'_{\dot{v}})(I'_z - N'_{\dot{r}}) - (m'x'_G - Y'_{\dot{r}})(m'x'_G - N'_{\dot{v}})$.

3.3. Model refinement

3.3.1. Overview

A maneuvering model is not automatically built by providing data. For any approach, it is integral to involve experts. Especially in this field, only a limited dataset is available for model development and validation due to the prohibitive cost of the data collection. Thereby, experts have been playing key roles to certify models based on their confidence in its performance (Berg, 2015). Their experience and knowledge characterize such confidence. Figs. 1 and 2 show a schematic comparison of the traditional data-based and proposed frameworks.

In the traditional approach, experts involve their knowledge mostly in the design of experiments, models, and identification algorithms. It requires much cost, time, and expertise to conduct them carefully. Then, hyperparameters of the identification algorithm are tuned. The PI is performed by using data. Experts at this moment judge the model's validity and check its performance for the selected maneuvers (The International Towing Tank Conference, 2002). Such evaluations are rather passive. It may result in carefully investigating, improving, or redoing some processes if they are less confident in the model's validity. This iteration seems inefficient when we have confidence in the similarity between resulting models of two similar ships.

On the contrary, by building up models on top of the source parameters, the proposed framework actively employs knowledge coming from past experience for similar ships as a modeling foundation. Firstly, experts need to agree that it is reasonable that the target model is derived by refining the source model in light of their domain knowledge and experience. This agreement allows the source model to act as a source of modeling confidence. Then, we highly constrain parameters towards the source parameters by defining conservative ranges of hyperparameters to do so. It could occur that we do not find parameters satisfying improvement criteria with defined ranges of hyperparameters. Then, the ranges are broadened in the next iteration leading to less prioritized the source parameters and more dependency on data. Hence, the proposed framework parsimoniously finds a model by gradually expanding the tuning ranges starting from the source parameters in a parameter space.

Such a practice has practical advantages. Through the proposed framework, we clearly understand the impact of updating the source parameters on reducing the empirical risk in the validation dataset. For example, in the first iteration, a model slightly adjusted from the source model but with moderate empirical risk could be found. In the next iteration, by less prioritizing the source parameters, we could find a model with very different parameters from the source parameters and small empirical risk. It is impossible to ensure which model has better generalization performance mathematically. This framework does not aim to do so. Rather, this study supports decisionmakers in selecting a model from a cloud of models by offering both options. Although it is not guaranteed there exist parameters satisfying the performance criteria within such a safe zone, this framework helps designers find them if they exist without being too sensitive to the data design, model design, and PI algorithm design by having a knowledge safeguard.

3.3.2. Source ship

In this study, we focus on the risk of global parameter search without a specific knowledge anchor in the traditional framework. The proposed framework, in contrast, performs local parameter search using a reliable similar-ship model as a knowledge foundation. As it is not practical to comprehensively verify model performance, we must select a high-performing model that can be justified from the perspective of domain knowledge. Introducing a reliable knowledge foundation into modeling can reduce the side effects of being extremely dependent on data in the traditional framework. On the other hand, using a dissimilar ship as a knowledge foundation and refining it to fit the target ship's maneuver cannot be justified from the perspective of domain knowledge. As this can lead to a loss of model reliability, it is necessary that the selection of the source ship needs to be justified and agreed upon by experts. Such an agreement is derived from their domain knowledge, experience, and how much risk they take in the project. In this study, we propose a three-step procedure to check the eligibility of the source ship such that experts have sufficient confidence in the similarity of maneuverability of the two ships.

- First, two ships must belong to the same ship category. A large body of previous works has built regression models for estimating hydrodynamic parameters from ship specifications. Such a similarity is defined only for ships within the same category. For instance, high-speed passenger ships and oil tankers have very different systems. The similarity in maneuverability between such highly divergent ships is rarely discussed.
- Second, the similarity of the two ships' maneuverability needs to be examined in light of domain knowledge in ship design. Kose et al. (1992) focused on the MPR-series ships and considered C_b, L/B, B/d, l_{cb}, stern type, rudder type, and bow type as the main parameters characterizing such ships. Yoshimura and Masumoto (2012) investigated the maneuverability of high-speed merchant ships and fishing vessels. They revealed that hydrodynamic coefficients related to a ship hull were highly determined by trim/d, C_b, L/B, and d/L. Mei et al. (2019) defined a similarity vector using ship specifications based on such knowledge in ship design. They can be criteria for checking the similarity of maneuverability between two ships. In this study, inspired by Mei et al. (2019), the similarity is examined by evaluating the similarity vector of ship maneuverability in (21).
- Third, designers must check the performance of source parameters when being applied to the collected dataset of maneuvers of the target ship. Predicted trajectories must capture a trend of true trajectories of the target ship in the collected dataset. Otherwise, it is hard for experts to have sufficient confidence in developing a target model only by adjusting source parameters.

where C_b is the block coefficient, L is the ship length, B is the ship breadth, d is the ship draft, and l_{cb} is the position of the center of buoyancy.

3.3.3. Algorithm

The linear regression model has a general form:

$$\hat{\boldsymbol{y}} = \hat{\beta}_0 + \sum_{j=1}^{p} \hat{\beta}_j \boldsymbol{x}_j \tag{14}$$

where $\hat{\boldsymbol{y}}_{1\times m}$ denotes the output vector, $\boldsymbol{X} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_p]_{p\times m}$ is the input matrix, $\hat{\beta}_0$ is the intercept, $\hat{\boldsymbol{\beta}} = [\hat{\beta}_1, \ldots, \hat{\beta}_p]_{1\times p}$ is the parameter vector of the regression model, m is the number of data samples, and p is the number of the input dimension. We assume the parameter vector of the source model $\boldsymbol{\beta}^{s\prime}$ (the source parameters) is available and that for the target ship $\hat{\boldsymbol{\beta}}'$ (the target parameters) needs to be identified. Note that the Abkowitz model takes $\hat{\beta}_0 = 0$. If some parameters are removed from (4)–(6) in the source model, such parameters are not included in (14). Usually, the input variable $\boldsymbol{x}_j$ is z-score normalized so that the penalization of $\hat{\boldsymbol{\beta}}$ is applied in the same scaling over the input variables as:

$$\boldsymbol{x}'_j = \frac{\boldsymbol{x}_j - \mu_j}{\sigma_j} \tag{15}$$

where μ_j and σ_j are the mean and standard deviation of the input variable in the training dataset. Then, we define $\boldsymbol{X}' = [\boldsymbol{x}'_1, \ldots, \boldsymbol{x}'_j, \ldots, \boldsymbol{x}'_p]_{p\times m} = [\boldsymbol{x}'_1, \ldots, \boldsymbol{x}'_i, \ldots, \boldsymbol{x}'_m]^T_{m\times p}$ as the z-score normalized input matrix. (14) is rewritten as:

$$\hat{\boldsymbol{y}} = \hat{\beta}'_0 + \sum_{j=1}^{p} \hat{\beta}'_j \boldsymbol{x}'_j \tag{16}$$

where we define:

$$\hat{\beta}'_j \equiv \hat{\beta}_j \sigma_j \tag{17}$$

$$\hat{\beta}'_0 \equiv \sum_{j=1}^{p} \hat{\beta}_j \mu_j + \hat{\beta}_0 \tag{18}$$

We estimate $\hat{\beta}'_0 = \sum_{i=1}^{m} y_i$ in the training dataset, where $\boldsymbol{y} = [y_1, \ldots, y_m]$ is the target vector in the training dataset. Under the assumption that the residuals between the target vector $\boldsymbol{y}$ and $\hat{\boldsymbol{y}}$ are normally distributed in addition to $\hat{\beta}_0 = 0$ in (7)–(9), this estimation is equivalent to the definition (18). The source parameters are converted as $\beta^{s\prime}_j \equiv \beta^s_j \sigma_j$ to fit this z-score normalization. Then, by using $\beta^{s\prime}_j$ as a knowledge foundation, the refined parameter vector $\hat{\beta}'_j$ for the target ship is formulated as:

$$\begin{aligned}\hat{\boldsymbol{\beta}}' = \arg\min_{\boldsymbol{\beta}'} & \frac{1}{m} \sum_{i=1}^{m} (\boldsymbol{\beta}'^T \boldsymbol{x}'_i - \Delta y_i)^2 \\ & + \lambda(\alpha \|\boldsymbol{\beta}'\|_2^2 + (1-\alpha)\|\boldsymbol{\beta}' - \boldsymbol{\beta}^{s\prime}\|_2^2)\end{aligned} \tag{19}$$

where $\lambda \geq 0$ is a hyperparameter for determining the regularization strength and $0 \leq \alpha \leq 1$ is a hyperparameter that balances the source-parameter vector $\boldsymbol{\beta}^{s\prime}$ and the zero vector in the regularization. $\Delta \boldsymbol{y} =$

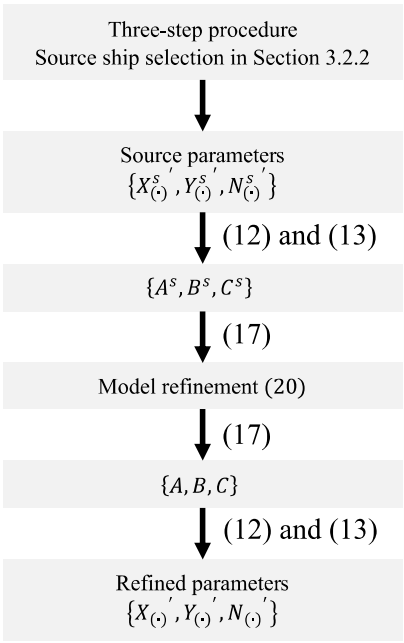


Fig. 3. An illustration for the procedure of model refinement.

$\boldsymbol{y} - \hat{\boldsymbol{\beta}}'_0$. A similar formulation for the l_1 regularization was presented in Takada and Fujisawa (2020) to handle continuously-changing environments without being too sensitive to noises and disturbances in the upcoming data. In this study, the source model represents a different system from that of the target ship, thereby, it is counterintuitive to constrain parameters towards the exact same values as the source parameters with the l_1 regularization. Thereby, inspired by a knowledge transfer with the l_2 regularization in Tommasi et al. (2010), this study expands it into the l_2 regularization. $\alpha = 1$ and $\lambda = 0$ correspond to the plain ridge regression and linear regression, respectively. Section 3.3.4 will explain how to determine these hyperparameters. (19) is equivalent to:

$$\hat{\boldsymbol{\beta}}' = (\boldsymbol{X}'^T \boldsymbol{X}' + m\lambda \boldsymbol{I})^{-1}(\boldsymbol{X}'^T \boldsymbol{\Delta y} + m\lambda(1-\alpha)\boldsymbol{\beta}^{s\prime}) \tag{20}$$

thereby, $\hat{\boldsymbol{\beta}}'$ is derived analytically. Based on the procedure shown in Fig. 3, this study forwards source parameters $\{X^{s\prime}_{(\cdot)}, Y^{s\prime}_{(\cdot)}, N^{s\prime}_{(\cdot)}\}$ into the target model. The non-dimensional hydrodynamic derivatives $\{X^{s\prime}_{(\cdot)}, Y^{s\prime}_{(\cdot)}, N^{s\prime}_{(\cdot)}\}$ are converted into $\{A^s, B^s, C^s\}$ corresponding to the regression forms in (7)–(9) by using (12) and (13). They are normalized with (17). Then, (20) derives refined parameters for the target ship. Finally, non-dimensional hydrodynamic derivatives for the target ship $\{X'_{(\cdot)}, Y'_{(\cdot)}, N'_{(\cdot)}\}$ is derived by reconverting refined parameters through (12) and (13).

3.3.4. Hyperparameter tuning

In (19), α and λ are hyperparameters that characterize the l_2 regression terms. As mentioned, ranges of hyperparameter tuning are defined by experts to balance their confidence in knowledge against data-driven calibration. In such ranges, they are tuned so that it minimizes the Root Mean Squared Error (RMSE) in the validation dataset, which stays away from the training dataset. The Tree-structured Parzen Estimator optimizer (Bergstra et al., 2011), which is one of the Bayesian optimization methods, is employed in the hyperparameter optimization framework *optuna* (Akiba et al., 2019). After determining hyperparameters, the model refinement is performed for a whole dataset with those hyperparameters.

4. Case study

In the case study, a simulation experiment transferring knowledge between two ships was conducted to present an example case where the source model was helpful to find a robust-and-accurate model for the target ship. The case study performed the model refinement of the Abkowitz model of a 161m-length container Mariner class vessel (Chislett and Strom-Tejsen, 1965), which was referred to as a source ship, for better fitting maneuverings of a 175m-length container ship SR108 (Son and Nomoto, 1981), which was referred to as a target ship. For the source ship, an Abkowitz model has been widely-used and validated. It is a source model and its parameters are source parameters in the case study. For the target ship, we assumed only a limited dataset was available. The source and target ships were highly similar, however, we found that the source parameters had room to be calibrated for maneuvers of the target ship. In the case study, we aimed to reduce 50% of the single-step-ahead velocity prediction error in the validation dataset in three directions. In practice, such criteria need to be carefully agreed upon with experts.

4.1. Source and target ships

A comparison of basic specifications of source and target ships are shown in Table 1. The source model for the source ship was for the operation speed of 15.0 knots. To evaluate the similarity between the two ships, inspired by Mei et al. (2019), a characteristic vector l is defined based on domain understanding of ship design:

$$l = [C_b, L_{pp}/B, B/T, L_{pp}/\Delta^{1/3}, Ar/(L_{pp}T), D_p/T] \tag{21}$$

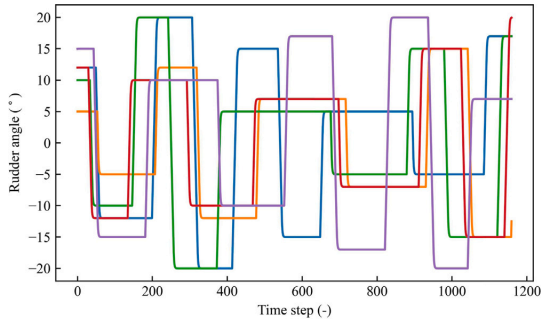


Fig. 4. Time series of the rudder angle during zigzag maneuverings for five training datasets in the case study with low-speed maneuverings of the target ship.

Table 1
Specifications of two ships.

	Source ship	Target ship
Ship name	Mariner (Chislett and Strom-Tejsen, 1965)	SR108 (Son and Nomoto, 1981)
Length between perpendiculars	160.9 m	175.0 m
Breadth	23.2 m	25.4 m
Design draft	8.2 m	8.5 m
Design displacement	18541 m^3	21222 m^3
Propeller diameter	6.706 m	6.533 m
The number of propellers	1	1
The number of blades	4	5

where C_b is the block coefficient, L_{pp} is the length between perpendiculars, T is the draught, Δ is the displacement, Ar is the rudder area, and D_p is the diameter of the propeller. $l^s = [0.59, 6.94, 2.82, 6.08, 0.023, 0.81]$ was for the source ship and $l^t = [0.56, 6.89, 2.99, 6.32, 0.022, 0.77]$ was for the target ship. Hence, two ships, which were categorized in the same ship type, have very similar characteristic vectors. Based on domain knowledge, quite similar parameters are to be estimated for the source and target ships. In fact, using the source parameters for maneuvers of the target ship reproduced the trend of the target ship's response well, however, a data-driven calibration was still necessary for better performance.

4.2. Dataset

On the MSS simulator (Perez and Fossen, 2009), multiple maneuvers of the target ship were conducted for refining the source parameters in the case study. The MSS is a widely-used open-source simulator developed by the Norwegian University of Science and Technology. The mathematical model of the target ship on the MSS simulator was a 4DoF nonlinear model (Son and Nomoto, 1981). A zigzag maneuver, 12.5° and −12.5° turning circle maneuvers were generated. They are standard maneuvers in the International Towing Tank Conference (ITTC) full-scale experiment (The International Towing Tank Conference, 2002). Each maneuver had a 650 s time series of positions, heading, velocities, and rudder angles saved in 2 Hz. To check if the proposed framework finds a robust-and-accurate model for different types of maneuvers of the target ship, three different case studies were conducted by taking three operation speeds $U_0 = 16.3$, 20.4, and 24.7 knots for low, middle, and high-speed maneuvering case studies, respectively. To evaluate the sensitivity of performances against having different training datasets, five training datasets were prepared for each case study by preparing five different zigzag maneuvers, of which time series of rudder angles are shown in Fig. 4 for the case study with low-speed maneuverings. In total, three case studies (high, middle, low-speed maneuvers of the target ship) were conducted, and each case

study had five different datasets for training and validation. The final evaluation was performed in a test dataset, which included 7.5° and −7.5° turning circle maneuvers. Maneuvers with 7.5° and −7.5° rudder angles were conducted only in the test dataset. Maneuvers in the test dataset were conducted at the corresponding high, middle, and low speeds for each case study. During the hyperparameter tuning, 70% of samples were employed for the training and the remaining samples were kept untouched for validation. The PI was performed in three directions independently. They were used only for the final evaluation.

4.2.1. Disturbance set up

Saved signals of the surge, sway, and yaw velocities were contaminated with the process noise $\boldsymbol{w}_i$ to represent the measurement noise in the real-world experiment. Inspired by Sutulo and Guedes Soares (2014), it was defined as:

$$\boldsymbol{w}_i = A_i k^0 k_i \boldsymbol{\xi} \tag{22}$$

where A_i is the reference amplitude of noises (8.0 m/s, 1.0 m/s, and 1.0 °/s for the surge, sway, and yaw velocities, respectively), k_i is a response-specific reduction factor (0.1, 1.0, and 1.0 for the surge, sway, and yaw velocities, respectively), $k^0 = 0.3\%$ is the general reduction factor. $\boldsymbol{\xi}$ denotes a raw signal. The Savitzky–Golay filter was applied to smooth contaminated signals before the PI. Then, 1160 time steps right after turning the rudder were extracted for the case study.

4.3. Baseline algorithms

To evaluate the benefit of the proposed method, four baseline PI algorithms were implemented. In addition, a prediction made with the source parameters without data-driven calibration was also conducted. They were implemented in the scikit-learn library (Pedregosa et al., 2011) in Python. Hyperparameter optimization was conducted for all baselines so that it minimized the empirical risk as much as possible. Regularization strengths were tuned in the log domain. The same abbreviations for algorithms were used in tables and figures in this section. Results for the present framework are represented by the abbreviation "MR" hereinafter. Note that RR, SVR, LR, and EN are baselines that do not employ a knowledge foundation, which are referred to as data-based frameworks in this section.

4.3.1. Source parameters (SP)

SP indicates the predictions for maneuvers of the target ship only by copying source parameters without any data-driven updates.

4.3.2. Ridge regression (RR)

RR is a linear regression with the l_2 regularization to avoid identifying parameters overfitting the training dataset. l_2 regularization strength $\alpha \in [10^{-10}, 10^{10}]$ was tuned.

4.3.3. Linear support vector regression (SVR)

SVR minimizes the loss function with the epsilon-insensitive zone, thus being robust to noise. The linear SVR is one of the most widely-used algorithms for the PI of ship dynamic models. The loss function with the l_2 loss was selected. The epsilon parameter was set to zero. The regularization parameter $C \in [10^{-10}, 10^{4}]$ was tuned.

4.3.4. Lasso regression (LR)

A feature selection is important to enhance the model's robustness. LR is a linear regression with l_1 regularization, which performs a regularization and feature selection. l_1 regularization strength $\alpha \in [10^{-10}, 10^{10}]$ was tuned.

4.3.5. Elastic net (EN)

EN is a linear regression with a combination of l_1 and l_2 regularizations. The penalty term $\alpha \in [10^{-10}, 10^{10}]$ and the ratio of the l_1 reguralization $l_1 \in [0.0, 1.0]$ were tuned.

Table 2
Parameters identified by the present MR, SP, and four baseline algorithms for the surge direction ($\times 10^{-5}$) by using one of the five training datasets. Results here are for the same case as Fig. 5 shows.

	SP	**MR**	RR	LR	EN	SVR
X'_u	−184	−177	−9	0	0	−14
X'_{uu}	−110	−82	22	0	0	35
X'_{uuu}	−215	−264	−62	0	0	−108
X'_{vv}	−899	−977	−157	0	0	−236
X'_{rr}	18	−38	−50	−99	−78	−73
$X'_{\delta\delta}$	−95	−143	−31	0	0	−50
$X'_{u\delta\delta}$	−190	−100	−52	0	0	−81
X'_{rv}	798	826	88	0	0	129
$X'_{v\delta}$	−93	−20	73	0	0	100
$X'_{uv\delta}$	−93	−51	22	0	0	61
X'_{uvv}	0	0	−102	0	0	−164
X'_{urr}	0	0	−16	0	0	−31
X'_{uvr}	0	0	36	0	0	64
$X'_{r\delta}$	0	0	−42	0	0	−58
$X'_{ur\delta}$	0	0	−23	0	0	−51
X'_0	0	0	−4	0	0	−6

Table 3
Parameters identified by the present MR, SP, and four baseline algorithms for the sway direction ($\times 10^{-5}$) by using one of the five training datasets. Results here are for the same case as Fig. 5 shows.

	SP	**MR**	RR	LR	EN	SVR
Y'_v	−1160	−1128	−253	−1300	−1300	−311
Y'_r	−499	−507	−131	−570	−570	−129
Y'_{vvv}	−8078	−5866	−12294	−21246	−21233	−14506
Y'_{vvr}	15356	14491	3925	484	494	3807
Y'_{vu}	−1160	−1294	649	120	119	722
Y'_{ru}	−499	−412	−21	192	191	46
Y'_{δ}	−278	−309	−296	−292	−292	−351
$Y'_{\delta\delta\delta}$	90	−251	−1044	1118	1117	−414
$Y'_{u\delta}$	−556	−427	432	−70	−68	416
$Y'_{uu\delta}$	−278	−713	−358	−185	−184	−510
$Y'_{v\delta\delta}$	−4	286	−1536	−7079	−7074	−2515
$Y'_{vv\delta}$	−1190	−2852	3611	−11034	−11000	5294
Y'_{0u}	−8	−2	26	−8	−8	17
Y'_{0uu}	−4	−13	97	−9	−9	77
Y'_0	−4	−2	12	1	1	10
Y'_{rrr}	0	0	−116	225	221	−597
Y'_{vrr}	0	0	−727	−2355	−2353	−366
Y'_{vuu}	0	0	−383	220	221	−615
Y'_{ruu}	0	0	−181	0	0	−291
$Y'_{r\delta\delta}$	0	0	−226	−3117	−3114	−659
$Y'_{rr\delta}$	0	0	174	390	394	96
$Y'_{rv\delta}$	0	0	−1072	−494	−481	−1932

4.4. Results

For three directions, firstly, tuning ranges of hyperparameters were set to $\alpha \in [0.0, 0.1]$ and $\lambda \in [10, 10^{10}]$ that highly constrain parameters towards the main parameters. Results with the tuned hyperparameters are shown hereinafter. Fig. 5 show RMSEs of single-step-ahead velocity prediction in the surge, sway, and yaw directions in the validation dataset. In Fig. 5, results using one of the training datasets of low-speed maneuverings are shown. Tables 2–4 show parameters refined by the present MR in three directions with SP and those made by four baseline algorithms. Abbreviations in figures and tables are defined in Section 4.3. In the surge direction, having a knowledge foundation in MR was found to be powerful in reducing validation loss. As shown in Fig. 5, whereas MR performed the best in the validation dataset with highly constrained parameter change from SP, data-based frameworks (RR, SVR, LR, and EN) resulted in much poorer performances with notably deviated parameters from SP. This result presented a challenge in solely relying on data in the case study since the dataset had limited excitation in the surge direction. In the sway direction, MR achieved an equivalent performance with the much less number of parameters, of which change was highly constrained towards SP, than

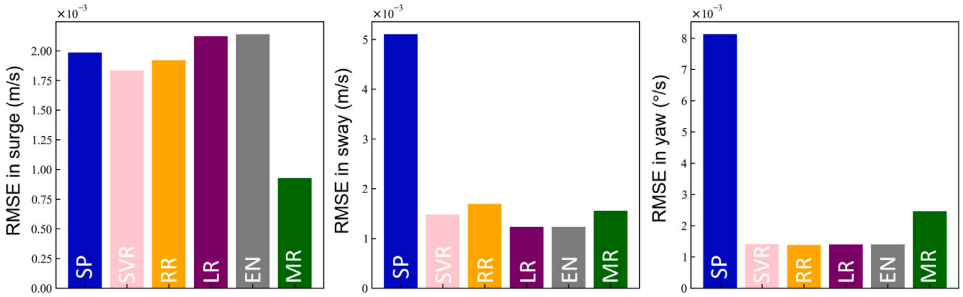


Fig. 5. RMSEs in the single-step-ahead velocity prediction in the surge, sway, and yaw directions made by the present MR, SP, and four baseline algorithms. Results here are for the low-speed case study with one of the five training datasets. Note that five lines with five different training datasets for MR highly overlap.

Table 4
Parameters identified by the present MR, SP, and four baseline algorithms for the yaw direction ($\times 10^{-5}$) by using one of the five training datasets. Results here are for the same case as Fig. 5 shows.

	SP	**MR**	RR	LR	EN	SVR
N'_{v}	−264	−243	−111	−88	−88	2
N'_{r}	−166	−150	−117	−100	−100	−48
N'_{vvv}	1636	667	−791	−682	−682	−2015
N'_{vvr}	−5483	−4545	−512	−1374	−1374	−797
N'_{vu}	−264	−187	300	229	230	188
N'_{ru}	−166	−176	−23	45	45	31
N'_{δ}	139	127	112	101	101	80
$N'_{\delta\delta\delta}$	−45	−15	−516	−473	−473	−416
$N'_{u\delta}$	278	207	135	17	17	42
$N'_{uu\delta}$	139	279	618	29	30	266
$N'_{vv\delta}$	13	−20	2866	2670	2670	2399
$N'_{vv\delta}$	489	1031	2060	2883	2885	2724
N'_{0u}	6	2	−2	2	2	5
N'_{0uu}	3	8	9	11	11	26
N'_{0}	3	1	4	3	3	3
N'_{rrr}	0	0	−465	−885	−885	−739
N'_{vrr}	0	0	1256	61	61	1319
N'_{vuu}	0	0	492	65	65	16
N'_{ruu}	0	0	−267	0	0	−154
$N'_{r\delta\delta}$	0	0	809	689	689	574
$N'_{r\delta\delta}$	0	0	1039	1335	1335	1021
$N'_{rv\delta}$	0	0	−56	692	693	−797

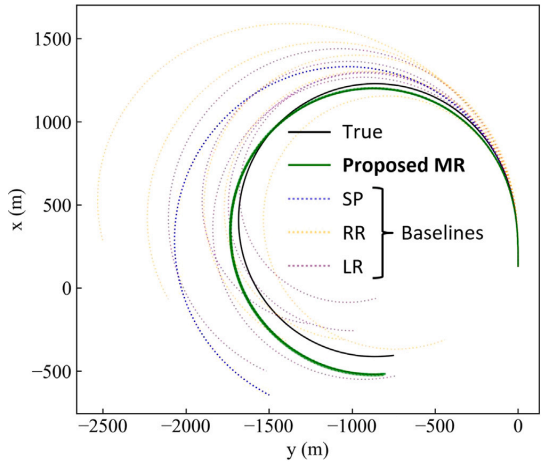


Fig. 6. True and predicted trajectories for the −7.5° turning maneuver in the test dataset. Trajectories made by EN and SVR are not shown for better visibility. Results here are for the same case as Fig. 5 shows. Note that five lines with five different training datasets for MR highly overlap.

that identified by the data-based frameworks. It can be seen as a big advantage since a simpler model is more robust than more complex models with equivalent performance. In the yaw direction, Fig. 5 shows a different trend from results in the other directions. MR calibrated SP well, however, models with very different parameters from the source parameters, which were built in data-based frameworks, performed better in the validation dataset than MR. Thereby, by broadening ranges of hyperparameter tuning and less prioritizing a knowledge foundation, the validation loss of MR could be reduced. Here is where experts introduce their confidence in knowledge against data-driven calibration. Given that a dataset was rather limited in this case study and MR achieved an improvement criterion of 50%, this case study did not allow models to further be calibrated by data by broadening ranges of hyperparameter tuning. As sufficient improvements were achieved by adjusting the source parameters, the same ranges of hyperparameter tuning were applied to all case studies. Hence, the proposed framework offered such a conservative option within the confidence range around the source parameters rather than forgetting it and building up a new model solely relying on data.

In Fig. 6, true and predicted trajectories for the −7.5° turning maneuver in the test dataset of the low-speed case study are shown. It should be noted that the test dataset was used only for the final evaluation. For better visibility, trajectories predicted by EN and SVR are not shown as they had a similar trend to those by LR and RR. Their performances will be listed in Table 5. A single prediction is shown for SP as it performs always the same without using any data. Five lines of predictions are shown for MR, LR, and RR since their performances fluctuate depending on which training dataset was employed. Fig. 7 shows true and predicted time histories of velocities of that maneuver. The two ships had similar characteristic vectors, thereby, it is discerned that the prediction made by SP captured the trend of the true trajectory and velocities' time histories although it had room for being refined for better performance. We see that data-based frameworks (LR and RR) resulted in poor performances, moreover, their performances highly fluctuated. It implies that data-based frameworks without a knowledge foundation have much risk of being trapped into deceptive models due to a jump between rival models in the clouds of almost-equally-accurate models. On the other hand, the present MR with a knowledge foundation performed more stably and accurately. It resulted in five predictions, shown in green, highly overlapping.

A summary of results is listed in Table 5. In the test maneuvers, this table compares prediction errors in four metrics defined by the ITTC for the turning-circle maneuvers: Steady Turning Radius (STR), Maximum Transfer (MT), Maximum Advance (MA), and Velocity Loss On Steady turn (VLS). For each metric, the error rate e (%) was calculated for all predictions.

$$e = \left| \frac{m_{\text{true}} - m_{\text{pred}}}{m_{\text{true}}} \right| \times 100 \tag{23}$$

Table 5
A comparison of performances of the present model refinement (MR) and five baseline algorithms for two turning circle maneuvers in the test dataset. Steady Turning Radius (STR), Maximum Transfer (MT), Maximum Advance (MA), and Velocity Loss on Steady turn (VLS) defined by the ITTC are compared. The average absolute error is that made by five different training datasets. The standard deviation of absolute error is its standard deviation. N/A indicates one or more trajectories made by five training datasets result in physically unreasonable trajectories. The red numbers show the best performance against the other frameworks for each metric of each maneuver.

			True	Average absolute error μ (%)						Standard deviation of absolute error σ (%)				
				SP	**MR**	LR	RR	EN	SVR	**MR**	LR	RR	EN	SVR
Case 1 Low speed	7.5° turning	STR (m)	824.9	10.3	3.6	12.7	N/A	11.8	N/A	0.4	6.3	N/A	6.5	N/A
		MT (m)	1683.2	6.1	4.5	5.4		5.6		0.4	5.4		5.0	
		MA (m)	1099.2	6.5	9.6	6.2		6.3		0.3	3.2		3.1	
		VLS (m/s)	1.3	38.2	0.9	51.1		56.2		0.6	19.1		10.0	
	−7.5° turning	STR (m)	817.1	28.0	5.4	16.5	10.7	16.5	17.6	0.2	5.1	7.5	5.9	7.1
		MT (m)	1683.2	22.9	2.9	10.3	21.9	10.9	12.3	0.2	8.6	17.2	8.5	9.3
		MA (m)	1099.2	9.3	2.6	9.4	16.9	9.6	14.0	0.2	5.7	10.0	5.7	9.6
		VLS (m/s)	1.3	49.5	9.0	56.7	45.9	60.9	64.8	0.8	9.9	4.4	5.2	7.3
Case 2 Middle speed	7.5° turning	STR (m)	760.7	19.6	1.0	25.8	32.1	24.8	N/A	0.3	15.2	14.2	13.1	N/A
		MT (m)	1558.5	14.6	1.8	19.2	30.7	18.8		0.3	6.4	14.0	5.7	
		MA (m)	1039.8	1.9	8.3	10.3	16.4	10.2		0.2	7.8	7.4	7.4	
		VLS (m/s)	1.8	43.8	2.3	43.2	26.0	29.6		0.6	35.0	11.1	12.8	
	−7.5° turning	STR (m)	755.3	38.5	7.0	14.5	N/A	13.6	27.1	0.4	4.7	N/A	4.1	18.1
		MT (m)	1558.5	32.8	4.7	15.1		14.9	25.3	0.5	4.2		4.2	29.6
		MA (m)	1039.8	14.4	2.5	9.2		9.0	22.4	0.5	2.6		2.5	15.5
		VLS (m/s)	1.8	54.1	10.4	51.3		34.1	55.3	0.9	31.7		15.0	12.2
Case 3 High speed	7.5° turning	STR (m)	680.0	33.8	1.1	14.0	77.9	14.0	N/A	0.3	6.3	60.6	5.1	N/A
		MT (m)	1375.3	29.8	2.4	11.9	39.1	11.8		0.3	6.9	29.8	7.2	
		MA (m)	962.3	5.1	7.3	6.5	8.1	6.3		0.2	2.5	5.2	2.1	
		VLS (m/s)	2.5	50.9	3.3	65.3	24.9	56.5		0.8	22.2	17.2	12.1	
	−7.5° turning	STR (m)	676.3	54.6	9.0	12.4	41.9	12.9	N/A	0.4	6.0	25.9	5.5	N/A
		MT (m)	1375.3	50.4	8.8	11.3	34.0	11.8		0.4	7.3	20.8	6.8	
		MA (m)	962.3	22.6	1.8	5.3	11.6	5.5		0.3	2.5	6.1	2.6	
		VLS (m/s)	2.5	60.0	10.6	64.5	27.8	56.0		0.9	22.8	13.9	12.1	

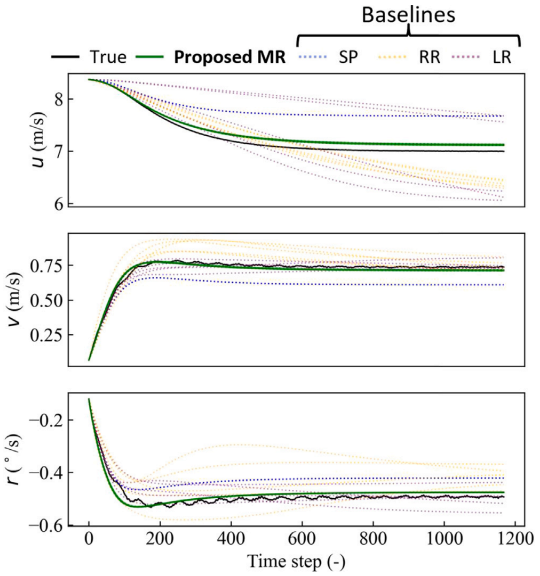


Fig. 7. True and predicted time histories of velocities for the −7.5° turning maneuver in the test dataset.Time histories made by EN and SVR are not shown for better visibility. Results here are for the same case as Fig. 5 shows.

where m_{true} and m_{pred} are true and predicted values for each metric. For each case study, five different training datasets were used. In Table 5, values in the columns for the average absolute error show μ for each metric:

$$\mu = \frac{1}{K}\Sigma_{i=1}^{K} e_i \tag{24}$$

where $K = 5$ is the number of the training datasets and e_i is e for prediction made by the ith training dataset. Please note that $\mu = e$ for SP since it does not use any dataset. Values in the columns for the standard deviation of absolute error show σ for each metric:

$$\sigma = \sqrt{\frac{1}{K}\Sigma_{i=1}^{K}(e_i - \mu)^2} \tag{25}$$

Note that $\sigma = 0$ for SP since it does not use any dataset, thereby, it is not shown in the table. Small μ and σ show that corresponding algorithms achieve the average good performance over experiments using five different training dataset and such a performance is stably accomplished. "N/A" indicates that one or more models in the five models identified by using five training datasets ended up with physically unreasonable trajectories. If the heading does not either monotonically decrease or increase, the trajectory was considered physically unreasonable. First, we focus on the average absolute errors. In terms of almost all metrics of almost all test maneuvers, SP made much prediction error. In particular, when the target maneuver was high-speed, their prediction errors were found to be large since the source parameters were not well-adjusted to it. On the other hand, such large prediction errors were notably mitigated in the present MR. As RR and SVR did not perform a feature selection, the identified models were not robust. LR and EN selected only informative features with the l_1 loss function, thereby, the identified models were robust and stable without having redundant features in the regression models. However, the present MR outperformed SP and four baseline algorithms in almost all metrics of almost all test maneuvers. In light of the standard deviation of the absolute errors, the robustness of the performance of the present MR against having different training datasets were clearly shown. It is reasonable since MR appreciates the source model and parsimoniously exploits the target datasets. Four baseline algorithms were found to be very sensitive to the selection of training datasets as they were fully dependent on data without having the confidence range around the source parameters.

4.5. Discussion

In the case study, the PI of the Abkowitz model for the target ship was conducted. Due to the high similarity between the ship characteristic vectors of the two ships, their resulting models were expected to be highly relevant. However, data-driven refinement of the source parameters was necessary to satisfy the performance criteria for maneuvers of the target ship. In the case study, the proposed framework was found to be powerful in easily finding a refined model with good

robustness and generalization performance, by prioritizing a safe zone of calibration derived by experts' confidence in parameter transfer from the source ship. We prepared five different training datasets in the case study. The proposed framework found a robust-and-accurate model and stably accomplished a great performance against having five different training datasets. On the contrary, without having the parameters of the source ship, model performance was found to highly fluctuate.

In the case study, the proposed framework showed better performance in the test dataset than traditional data-based frameworks. This result presented an example case where the proposed framework was powerful in easily finding a robust-and-accurate model within a safe zone drawn based on the source parameters without being sensitive to the experiment design. Nevertheless, it does not show an absolute limitation of data-based frameworks. Their performance could be improved by devoting more effort for the data design, model design, PI algorithm design. However, this study focuses on cases where we have less room to do so and need to efficiently deliver the target model. It would perform a complementary role to data-based frameworks in such cases rather than always replacing them. Especially for delivering models integrated into safety-critical systems, data-based frameworks remain important since collecting much data is the only way to get closer to the global optimum parameters.

Typically, shipyards build many ships with similar specifications, so it is expected that similar ship models may be found. Moreover, when constructing models for different loading and water-depth conditions, it is possible to consider a standard-condition model as a similar-ship model. Therefore, the scope of application for this framework is considered to be wide. On the other hand, it does not fully replace the traditional framework. When there is no reliable similar-ship model, the proposed framework cannot be used. To construct accurate and reliable models, either a knowledge foundation or data collected through careful experiments is required.

In this field, a validation study cannot be conducted comprehensively. The ITTC 29th Manoeuvring Committee (The International Towing Tank Conference, 2014) asked different institutes how they validate simulation models. A majority of answers were validations based on previous experience with other ships or benchmark data, expert judgment, and comparing the trajectories with free-running tests. Any of them does not have unified criteria. In practice, experts make an agreement so that they satisfy their confidence in the model's validity. Thereby, it is reasonable that the proposed framework introduces their decision into selecting a final model from rival models based on their understanding of the relevance of similar ships' dynamics. This study helps such a practice be efficiently conducted. As a matter of fact, it contributed to building a robust-and-accurate model for the target ship. Such a decision is not randomly made but based on experts' experience and how much risk the project takes.

In practice, we may have further knowledge that some parameters are not to be constrained towards the source parameters. For example, if a ship experiences a renovation of a propulsion system, related parameters need to be less penalized to be changed. Our future work will be devoted to investigating a positive impact of having such knowledge in the regularization term to better transferring knowledge between tasks.

In this study, we addressed the risk of conducting a global parameter search without any knowledge anchor. By using a reliable similar ship model to locally update the model, the knowledge connection between ships can be established and the risk can be reduced. To demonstrate this concept, this study dealt with a 3DoF maneuvering model in calm water. However, nowadays, models that include external-force models such as waves (e.g., Yasukawa et al., 2021) are considered more practical. The applicability of the proposed framework to such advanced models has not been demonstrated in this paper. However, given that the external force term is a parametric model that relates to a ship's mechanical performance, this framework is also expected to contribute to reducing the number of experiments and risks associated with creating an external-force model. On the other hand, defining similarities between ships becomes more complicated in models that consider external forces. For example, even if the maneuvering characteristic is similar, there may be a significant deviation in the dynamic positioning characteristic. Regarding the specific definition of similarity, further discussion will be necessary in our future study.

5. Conclusion

When building a new maneuvering model for a new ship, this article proposed a new framework that employs parameters of a maneuvering model for a similar ship as a safeguard in the system identification approach. The proposed framework introduces parameters of such a model as the l2 regularization in the ridge regression for identifying model parameters for the new ship. Its hyperparameters represent a regularization strength towards the knowledge foundation. By limiting ranges of hyperparameter tuning, experts introduce their confidence in such a knowledge foundation into a data-driven calibration. It helps designers deliver a robust-and-accurate model within the confidence range of data-driven calibration efficiently, without being too sensitive to the design of experiments. The proposed framework bridges dynamics of two similar ships. It is effective for experts to comprehend its relevance and discrepancy for having sufficient confidence in the validity of the new model. In a simulation case study, a knowledge transfer between the Abkowitz models from 161 m to 175 m container ships was conducted. By using the model parameters of the former ship, a robust-and-accurate model for the latter ship was built only by adjusting the parameters of the former ship. It was achieved stably for having different training datasets. In the field of ship maneuvering, comprehensive data collection is unfeasible. Thereby, as the proposed framework introduced, it is of great importance for experts to actively introduce their confidence in the similarity of models for similar ships into data-driven calibration, thus enhancing the consistency with domain knowledge and data-driven calibration. It is helpful to readily deliver robust-and-accurate maneuvering models for similar ships in simulators. In practice, it becomes important to incorporate more experts' knowledge and experience into defining such a confident range. It must be discussed on a case-by-base basis, thereby, next research direction must include more case studies bridging real-life tasks.

CRediT authorship contribution statement

Motoyasu Kanazawa: Conceptualization, Methodology, Investigation, Writing – original draft. **Tongtong Wang:** Writing – review & editing. **Yasuo Ichinose:** Conceptualization, Writing – review & editing. **Robert Skulstad:** Supervision. **Guoyuan Li:** Writing – review & editing, Supervision. **Houxiang Zhang:** Writing – review & editing, Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgments

This work was supported by a grant from the Research Council of Norway, IKTPLUSS project No. 309323 "Remote Control Center for Autonomous Ship Support" in Norway.

References

Akiba, T., Sano, S., Yanase, T., Ohta, T., Koyama, M., 2019. Optuna: A next-generation hyperparameter optimization framework. In: KDD '19: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. pp. 2623–2631.

Alexandersson, M., Mao, W., Ringsberg, J., 2022. System identification of vessel manoeuvring models. Ocean Eng. 266, 112940, https://linkinghub.elsevier.com/retrieve/pii/S0029801822022235.

Berg, T., 2015. Memo D0-SIMVAL state of the art document on validation of simulation models for ship manoeuvring PERSON RESPONSIBLE / AUTHOR project partners X PROJECT NO / FILE CODE. www.marintek.sintef.no.

Bergstra, J., Bardenet, R., Bengio, Y., Balazs, K., 2011. Algorithms for hyper-parameter optimization. In: Advances in Neural Information Processing Systems. NIPS 2011, pp. 1–9.

Cardenas, P., De Barros, E., 2020. Estimation of AUV hydrodynamic coefficients using analytical and system identification approaches. IEEE J. Ocean. Eng. 45, 1157–1176.

Chislett, M., Strom-Tejsen, J., 1965. Planar motion mechanism tests and full-scale steering and manoeuvring predictions for a Mariner class vessel. Int. Shipbuild. Prog. 12, 201–224.

Fossen, T., 2011. Handbook of Marine Craft Hydrodynamics and Motion Control. Wiley.

Kose, K., Misiag, W., Zhu, J., Hirao, S., 1992. Database system approach for maneuvering performance prediction. J. Soc. Naval Archit. Japan 172, 375–382.

Ljungberg, F., Linder, J., Enqvist, M., Tervo, K., 2022. Experiment design for identification of marine models. http://arxiv.org/abs/2210.05496.

Luo, W., Guedes Soares, C., Zou, Z., 2016. Parameter identification of ship maneuvering model based on support vector machines and particle swarm optimization. J. Offshore Mech. Arctic Eng. 138, 1–8.

Luo, W., Li, X., 2017. Measures to diminish the parameter drift in the modeling of ship manoeuvring using system identification. Appl. Ocean Res. 67, 9–20. http://dx.doi.org/10.1016/j.apor.2017.06.008.

Luo, W., Zou, Z., 2009. Parametric identification of ship maneuvering models by using support vector machines. J. Ship Res. 53, 19–30.

M.A., A., 1964. Lectures on Ship Hydrodynamics - Steering and Maneuverability. Hydro-.

Martín-delosReyes, L., Jiménez-Mejías, E., Martínez-Ruiz, V., Moreno-Roldán, E., Molina-Soberanes, D., Lardelli-Claret, P., 2019. Efficacy of training with driving simulators in improving safety in young novice or learner drivers: A systematic review. Transp. Res. Part F: Traffic Psychol. Behav. 62, 58–65.

Mei, B., Sun, L., Shi, G., 2019. White-black-box hybrid model identification based on RM-RF for ship maneuvering. IEEE Access 7, 57691–57705.

Mucha, P., El Moctar, O., 2015. Revisiting mathematical models for manoeuvring prediction based on modified Taylor-series expansions. Ship Technol. Res. 62, 81–96.

Myers, P., Starr, A., Mullins, K., 2018. Flight simulator fidelity, training transfer, and the role of instructors in optimizing learning. Int. J. Aviat., Aeronaut., Aerosp. 5.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E., 2011. Scikit-learn: Machine learning in Python. J. Mach. Learn. Res. 12, 2825–2830.

Perez, T., Fossen, T., 2009. A Matlab toolbox for parametric identification of radiation-force models of ships and offshore structures. Model., Identif. Control 30, 1–15.

Sargent, R., 2010. Verification and validation of simulation models. In: Proceedings - Winter Simulation Conference. pp. 166–183.

Son, K., Nomoto, K., 1981. On the coupled motion of steering and rolling of a high speed container ship. Nav. Archit. Ocean Eng. 150, 73–83.

Sutulo, S., Guedes Soares, C., 2014. An algorithm for offline identification of ship manoeuvring mathematical models from free-running tests. Ocean Eng. 79, 10–25.

Takada, M., Fujisawa, H., 2020. Transfer learning via l1 regularization. In: 34th Conference on Neural Information Processing Systems. NeurIPS 2020, pp. 14266–14277.

The International Towing Tank Conference, 2002. The International Towing Tank Conference ITTC-Recommended Procedures: Full Scale Measurements Manoeuvrability Full Scale Manoeuvring Trials Procedure.

The International Towing Tank Conference, 2014. The International Towing Tank Conference ITTC Quality System Manual Recommended Procedures and Guidelines: Validation of Manoeuvring Simulation Models.

Tommasi, T., Orabona, F., Caputo, B., 2010. Safety in numbers: Learning categories from few examples with multi model knowledge transfer. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition. pp. 3081–3088.

Wang, Z., Guedes Soares, C., Zou, Z., 2020. Optimal design of excitation signal for identification of nonlinear ship manoeuvring model. Ocean Eng. 196, 106778. http://dx.doi.org/10.1016/j.oceaneng.2019.106778.

Wang, T., Li, G., Hatledal, L., Skulstad, R., Aesoy, V., Zhang, H., 2021a. Incorporating approximate dynamics into data-driven calibrator: A representative model for ship maneuvering prediction. IEEE Trans. Ind. Inform. 1.

Wang, T., Li, G., Wu, B., Æsøy, V., Zhang, H., 2021b. Parameter identification of ship manoeuvring model under disturbance using support vector machine method. Ships Offshore Struct. 16, 13–21.

Xu, H., Hassani, V., Guedes Soares, C., 2020. Truncated least square support vector machine for parameter estimation of a nonlinear manoeuvring model based on PMM tests. Appl. Ocean Res. 97.

Yasukawa, H., Hasnan, M., Matsuda, A., 2021. Validation of 6-DOF motion simulations for ship turning in regular waves. J. Mar. Sci. Technol. (Japan) 26, 1096–1111.

Yoshimura, Y., Masumoto, Y., 2012. Hydrodynamic database and manoeuvring prediction method with medium high-speed merchant ships and fishing vessels. In: Marine Simulation and Ship Manoeuvrability International Conference.

Zhu, M., Wen, Y., Sun, W., Wu, B., 2019. A novel adaptive weighted least square support vector regression algorithm-based identification of the ship dynamic model. IEEE Access 7, 128910–128924.

F

Paper VI

This paper is awaiting publication and is not included in NTNU Open

ISBN 978-82-326-7194-6 (printed ver.)
ISBN 978-82-326-7193-9 (electronic ver.)
ISSN 1503-8181 (printed ver.)
ISSN 2703-8084 (online ver.)

NTNU
Norwegian University of
Science and Technology