Doctoral theses at NTNU, 2021:383

Robert Skulstad

Data-based modelling of ships

For motion prediction and control allocation

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

Robert Skulstad

Data-based modelling of ships

For motion prediction and control allocation

Thesis for the Degree of Philosophiae Doctor

Ålesund, September 2021

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 978-82-326-6093-3 (printed ver.)
ISBN 978-82-326-6605-8 (electronic ver.)
ISSN 1503-8181 (printed ver.)
ISSN 2703-8084 (online ver.)

Doctoral theses at NTNU, 2021:383

Printed by NTNU Grafisk senter

Abstract

Vessels operating on the surface of the ocean are exposed to an array of disturbances. These may come in terms of environmental disturbances, but may also come from signal loss. Modelling the behaviour of ships using physics-based models have therefore been pursued extensively to accommodate improved state estimation and motion control. These models map inherent ship motion states such as speed, heading, acceleration, wind effects and thruster forces from one discrete time step to the next (assuming equal measurement frequency). As such they are one-step predictors that enhance our understanding of the vessel motion.

With the increased focus on autonomous ships recently there is a growing need for technology that allows for extending, or projecting this window into the future. The aim of such a system is to supply a glimpse of the near-future trajectories to aid both the ship operator (autonomous ship operating system, or human ship operator) and operators at onshore remote control centres to get a better situational understanding. Ultimately this would lead to safer, and potentially more efficient, operations. The need for such a system is not ever-present. Thus a few scenarios were highlighted, which involve maneuvering in areas populated by other vessels (docking operation) and stationkeeping in proximity of fixed installations (power failure or position reference failure). To verify the benefit of including data-driven models for the aforementioned cases, ship simulators and historical data from the research vessel Gunnerus was applied. For the docking and power failure scenarios, the combined use of the identified maneuvering model and a neural network model was proposed. The, now hybrid, model was trained in a supervised fashion on relevant motion-related data. This lead to increased prediction performance compared to applying just the identified maneuvering model.

Low-speed maneuvering and stationkeeping are operational modes typically associated with ships that perform specialized tasks such as the research vessel Gunnerus that support the use of remotely operated vehicles and autonomous underwater vehicles. Other operations that require such operational modes include deploying subsea installations and pipeline installation and maintenance. To ensure safety and maneuvering capability the vessels used in these operations have redundancy in terms of thrusters and power generators. Having a redundant set of thrusters suggests that there are more than one set of thruster commands that may fulfill the overall motion controller request. The control allocation thus distributes thruster commands to ensure that the motion controller requests are met, but also that the commands honor inherent thruster constraints and leads to a minimized power consumption. The resulting optimization problem may be quite complex to solve because of the mentioned constraints. In this thesis a neural network was applied to yield an efficient evaluation of the mapping between the motion controller requests and individual thruster commands. To facilitate optimization within the bounds of the constraints of the system, custom loss functions were applied. They

provide a metric for the performance of the network and this is subsequently used to shape the response of the network during an initial offline training period. When training has completed the thruster commands may be obtained through a forward pass of the network using the motion controller request.

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 department and the Knowledge-Building Project for Industry "Digital Twins for Vessel Life Cycle Service" under Project 280703 and in part by a grant from the Research-Based Innovation "SFI Marine Operation in Virtual Environment," Norway, under Project 237929.

My supervisors throughout this Ph.D. project have been Prof. Houxiang Zhang, Prof. Thor I. Fossen and Dr. Bjørnar Vik. I would like to thank all of them for their support and feedback. In particular, I would like to thank my main supervisor Prof. Houxiang Zhang for his guidance and for being a constant source of motivation. Prof. Guoyuan Li, whom I had the pleasure of sharing an office with at the beginning of my Ph.D. period: Thank you for your help and insight regarding all aspects of my work. A big thank you to my colleagues at the Intelligent Systems Lab (formerly Mechatronics group) at NTNU Aalesund as well for our discussions. It has been a pleasure to be a part of the group. Especially, I would like to thank Dr. Lars Ivar Hatledal and Pierre Major for their help in issues related to simulation software.

Finally, to my partner Antonia and my family: Thank you for your patience and support.

Contents

List of Publications

This thesis is based on research resulting in four journal papers and one conference paper. They are all enclosed in the appendix section. In the following list of publications, the papers are listed chronologically by the date of initial submission, from the oldest one to the most recent. Note that Paper V has not yet been accepted for publication by the target journal.

I R. Skulstad, G. Li, T. I. Fossen and H. Zhang, "A Neural Network Approach to Control Allocation of Ships for Dynamic Positioning", *11th IFAC Conference on Control Applications in Marine Systems, Robotics, and Vehicles*, vol. 51, issue 29, pp. 128–133, 2018.

II 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 & Automation Magazine*, vol. 26, issue 3, pp. 39–51, 2019.

III 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, 2020.

IV R. Skulstad, G. Li, T. I. Fossen, T. Wang and H. Zhang, "A Co-operative Hybrid Model For Ship Motion Prediction", *Modeling, Identification and Control*, vol. 42, issue 1, pp. 17-26, 2021.

V R. Skulstad, G. Li, T. I. Fossen, B. Vik and H. Zhang, "Constrained Control Allocation For Dynamic Ship Positioning Using Deep Neural Network", *IEEE Robotics & Automation Magazine*, submitted, 2021.

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

i X. Cheng, G. Li, R. Skulstad, P. Major, S. Chen, H. Zhang and H. P. Hildre, "Data-driven uncertainty and sensitivity analysis for ship motion modeling in offshore operations", in *Ocean Engineering*, vol. 179, pp. 261-272, 2019.

ii X. Cheng, G. Li, R. Skulstad, H. Zhang and S. Chen, "SpectralSeaNet: Spectrogram and Convolutional Network-based Sea State Estimation", in *46th Annual Conference of the IEEE Industrial Electronics Society*, pp. 5069-5074, 2020.

iii L I. Hatledal, R. Skulstad, G. Li, A. Styve and H. Zhang, "Co-simulation as a Fundamental Technology for Twin Ships", in *Modeling, Identification and Control*, vol. 41, no. 4, pp. 297-311, 2020.

iv X. Cheng, G. Li, R. Skulstad, S. Chen, H. P. Hildre and H. Zhang "A Neural Network-Based Sensitivity Analysis Approach for Data-Driven Modeling of Ship Motion.", in *IEEE Journal of Oceanic Engineering*, vol. 45, no. 2, pp. 451-461, 2020.

v C. Wang, G. Li, R. Skulstad, X. Cheng, O. L. Osen and H. Zhang, "A sensitivity quantification approach to significance analysis of thrusters in dynamic positioning operations.", in *Ocean Engineering*, vol. 223, 2021.

List of Abbreviations

GNSS	Global Navigation Satellite Systems
RPM	Revolutions per Minute
ML	Machine Learning
SA	Situational awareness
DOF	Degree of freedom
RV	Research Vessel
OSC	Offshore Simulator Centre
OSP	Open Simulator Platform
FMU	Functional Mock-up Unit
ROV	Remotely Operated Vehicle
DP	Dynamic positioning
NED	North-East-Down
NN	Neural Network
AR	Autoregressive
RNN	Recurrent Neural Network
SVR	Support Vector Regression
QP	Quadratic Programming
SQP	Sequential Quadratic Programming
MPC	Model Predictive Control
MLP	Multi-Layered Perceptron
LSTM	Long Short-Term Memory
ReLU	Rectified Linear Unit
DNN	Deep Neural Network
DR	Dead reckoning

SLFN	Single-layer feedforward network
KF	Kalman filter

Nomenclature

F	Thrust from a propeller
$\boldsymbol{\alpha}$	Thruster azimuth angle vector
l	Length
$\boldsymbol{\eta}$	NED position and heading vector
$\boldsymbol{\nu}$	Linear and rotational velocity vector
ψ	Heading angle
$\boldsymbol{R}$	Rotation matrix
$\boldsymbol{M}$	Mass matrix
$\boldsymbol{C}$	Coriolis-centripetal matrix
$\boldsymbol{\tau}$	Force vector
$\boldsymbol{B}$	Thruster configuration matrix
$\boldsymbol{u}$	Thruster command vector
ρ_a	Density of air
$\boldsymbol{C}_{X/Y/N}$	Wind force coefficient matrices
V_w, V_{rw}	Wind velocity, velocity relative to the ship
β_w	Wind angle
γ_{rw}	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
o	Output signal of a NN node
$a()$	Activation function of a node in an MLP
$\boldsymbol{w}, \boldsymbol{W}$	Weight vector, matrix
$\boldsymbol{x}, \boldsymbol{X}$	General input vector, matrix
$\boldsymbol{b}$	Bias vector

$\boldsymbol{h}$	Hidden state vector
$\boldsymbol{c}$	Cell state vector
$\boldsymbol{y}$	General output vector
$L, Loss$	Loss value for NNs
λ, γ	Weighting factors
R	Function yielding NN complexity
$\boldsymbol{p}$	North-East position vector in NED
$\boldsymbol{v}$	Longitudinal and lateral velocity vector
t	Time
e	Position error

List of Figures

List of Tables

1 Introduction

With the growth of emerging demands from offshore applications, such as seabed survey, pipeline maintenance and offshore oil installations, the complexity of ship maneuvering during offshore operations, increases as more constraints from position accuracy, limited working space, and collision avoidance between vessels and floating and static structures, need to be taken into consideration. To assist to address the complexity and aid the ship operator, new knowledge and technology for such constrained ship maneuvering, are in demand.

On a maritime vessel - the term ship and vessel will both be used - sensory equipment are plentiful. Speed, orientation and positions may be derived from sensors such as Global Navigation Satellite Systems (GNSS) sensors, gyroscopes and accelerometers combined. The state of the propulsion system including power measurements, Revolutions per Minute (RPM), torque and propulsor/rudder angles are also sampled and used by motion control systems. The wind speed and direction are also sampled. In addition to being used by motion control systems, the sensor data is processed by the vessel operator and, based on experience, contributes to situational awareness (SA) [1]. SA is a key term in maritime operations, where safety is paramount. This applies to ships that are operated by a human and the case where ships are operated (semi) autonomously. The latter being a trend in recent years and considered by many to be an important step to increase efficiency and safety of shipping in the decades to come [2].

Data-driven modelling is becoming increasingly efficient and commonplace. It is an umbrella term, for which its sub-category, Machine Learning (ML), is often interchangeably used in its place. ML is a domain typically considered to hold the sub-categories supervised/unsupervised learning and reinforcement learning. Supervised learning, which is the learning strategy used in this thesis, may be applied when the data is structured and the effect of a set of input variables may be quantified in one or more output/target variables. As such the model learns by examples. ML thus provides for opportunities to enhance analyses and modelling of processes in a variety of domains, the maritime domain included. The basic pre-requisite is access to data originating from the process.

In this thesis ML contributes to two adjoined fields: motion prediction and control allocation. Both these fields typically involve low-speed maneuvering and as such they are engaged when the ship is operating in proximity to other ships or structures or performing seabed installation operations.

1.1 Data-based modelling

Data-based modelling of an object's motion has the potential to replace conventional first-principle-based modelling. In the maritime domain, especially for vessels maneu-

vering on the surface of the ocean, complex interactions with the ever-present wind, wave and current disturbances complicate the modelling problem. The various modes of operation, i.e. transit, low-speed maneuvering, stationkeeping [3], further adds to the challenge of producing a fixed model that may deliver a good approximation of the future motion of a ship, given measurable disturbances and actuation signals.

That being said, identification of parameters that go into established ship models is a mature field of research and well-established models are available for simulation and short-term propagation of ship motions [4]. However, for a prediction to be meaningful for applications besides e.g. offshore crane heave compensation [5] and maritime flight operations [6], the prediction horizon must be extended beyond a few seconds. These applications have a short time span due to fast-acting dynamics. The inertia of a ship, on the other hand, is typically large, which requires larger prediction horizons when operating in congested environments or close to fixed objects. In this thesis prediction horizons of 30-60 seconds have been applied in the case studies, where low-speed maneuvering close to a dock, and stationkeeping in open water were tested. Model-based motion prediction for ships moving at higher speeds is not considered in this thesis since these high velocities are normally associated with transit operations in non-congested waters.

Seeking to make use of the existing model knowledge leads to hybrid modelling [7]. In this context, the term hybrid is meant to reflect the merging of models from two domains: the kinetics/physics/first-principles-based domain and the data-based domain. In the kinetics-based domain, forces acting on the ship are modelled deterministically due to the interaction between the ship and its environment. Hydrodynamic and hydrostatic properties of the vessel, the shape of the hull, both below water (current-induced forces) and above (wind-induced forces), and the propulsors are key elements that dictate the overall dynamics in such a model. Wave-induced forces are typically discerned by applying an observer for closed-loop control scenarios or through vessel-specific response amplitude operators. In the data-based domain the effect of these elements are embedded in the model and learned based on the given input/target features and the data presented to it. The designer therefore has the opportunity of choosing what to represent with a data-based model. Various strategies for merging the information from the models exist. To mention a few; the parameters of the kinetic model may be adjusted by a data-based model [8], or the kinetic model may be applied as-is in cascade with a data-based model [9, 10] or in parallel [11, 12] to compensate the kinetic model error.

1.2 Ship motion prediction

Ship motion prediction is an enabling tool for operations involving ships and its immediate environment. It is mainly considered to act as a decision support tool, SA enhancer or collision detection tool and thereby augments the standard array of sensors by adding information about the likely future states of a ship. Depending on the application, the *future* may be a few seconds ahead of real time or several hours. However, ship motion prediction is generally considered to refer to the immediate future up to a few minutes. This limitation arises due to the complexity and uncertainty of the environment in which the vessel operates, limitations in terms of sensory equipment for wave propagation approaches [13] or the impact of the maneuvering actions of the vessel operator.

Existing ship motion prediction approaches leverage a wide variety of modelling techniques. Kinematic models enable the translation of acceleration and velocity measurements into positions. As such, they may be applied along with state estimators to project the estimate of these states into the future [14, 15]. Kinetic models, also termed dynamic models, rely on models that are parameterized through some kind of system identification approach. This is an involved process, which requires structured tests that allow proper parameter identification. An example of this approach was presented for the estimation of states related to aircraft takeoff and landing on a ship [16]. Modelling techniques that construct end-to-end models based only on data sampled from ships have received attention lately. These differ from the kinetic model approach due to lack of internal structure and physical relationship between the parameters of the model. These models are generally referred to as data-based or data-driven models.

Akin to the goal of aiding aircraft landing from [16], Yumori presented an autoregressive (AR) moving average model for predicting the heave motion of a vessel [17]. This approach utilized the time series of the vessel's heave displacement and the hydrodynamic pressure measured at the bottom of the vessel's hull to perform predictions of the heave displacement. As such, it is a good example of how data-based methods may accommodate any sensor type (water pressure) that may contribute information about the predicted state. Within data-based methods for prediction there are several model types that have been applied: Radial Basis Function networks [18, 19], evolutionary neural networks [20], recurrent neural networks (RNN) [21], Support Vector Regression (SVR) [22, 23, 24] and the AR model [25].

Longer term predictions of trajectories based on Automatic Identification System data, or non-time-domain predictions of single vessel states based on weather forecast, will not be covered in this thesis.

The motion prediction schemes presented in this paper, highlighted in Sections 2.3.1-2.3.3, has implications for manned ships as well as (semi) autonomous ships. In both cases the ship requires an operator (human, machine or remote human), which inevitably requires sufficient SA in order to conduct safe and efficient operations. The case studies given in Chapters 3 and 4 reflect operations that entail a certain amount of risk due to maneuvering close to other vessels or fixed structures. These case studies investigate three aspects of low speed maneuvering:

- Loss of position measurements
- Loss of power to control the ship
- Manual docking approaches

For all these cases the accuracy of the data/hybrid-based models was greater than the purely kinetic-based ones, which heightens the safety and efficiency of the operation.

1.3 Control allocation

Control allocation seeks to distribute force requests made by an overarching motion controller into commands individual to each thruster [26]. In doing so, the control allocation module should aim to generate the requested force as well as adhere to inherent constraints of the individual thrusters [27]. In some cases, constraints related to power

generators may also be considered [28, 29]. This field of research is prominent in both the aerospace and automotive domains, as well as the maritime domain. In the latter case control allocation is generally applied on over-actuated vessels that perform low-speed maneuvering and stationkeeping. Having more control inputs than the number of degrees of freedom (DOFs) that the motion controller controls (typically 3 DOF: surge, sway and yaw), means that the allocated command solution may not be unique. The use of rotatable thrusters, that may attain certain angles only slowly, leads to the general problem being nonlinear and non-convex, requiring significant computational resources [4]. State of the art methods locally approximate the problem using a convex Quadratic Programming (QP) approach in searching for solutions [30].

Apart from QP approaches to control allocation in the maritime domain, evolutionary search algorithms [31], population-based search algorithms [32] and filtering approaches [33] have been applied in the past. They offer other means of generating optimized control commands.

ML may be applied to this problem to obtain a fixed execution time and facilitate optimization in terms of inherent- and user-imposed constraints. A finite amount of parameters are adapted during the learning stage, which is where the bulk of the computation time is spent prior to running the trained allocator. The parameterized model can then be executed by a finite amount of computations.

1.4 Research questions

The majority of this thesis is concerned with ship motion prediction and how collected data can be used to enhance predictions and thereby provide ship operators with information relevant to conduct safer, more efficient operations. This leads to research questions one and two.

- **RQ1: Can data-based modelling provide predictions of sufficient length and quality to provide ship operators with safety-relevant information?**
- **RQ2: How can data-based modelling be applied/structured to provide motion prediction?**

In pursuing an answer to the above research question, an inherent issue related to key performance issues in data-based modelling appears. It has implications in terms of input dimension and training efficiency of ML models.

- **RQ3: Which input data/channels are important for data-based ship motion prediction?**

As models describing the dynamics of ships at sea have been researched extensively over the last decades, including such knowledge in a hybrid model seems beneficial. Approaches that combine the domains of ML and dynamic modelling of ships have therefore been pursued. Considering the extent of which the environmental disturbances can be measured, and thus included in the dynamic model, is also an important consideration at this point. It has implications for the structure of the hybrid model. This leads to the fourth research question.

- **RQ4: How can data-based modelling be combined with kinetic ship models to enhance the prediction performance?**

When performing predictions of the motion of an actuated vessel, future actuation signals may or may not be known. In the case of a manually operated vessel, the future intentions of the human operator is not known at the time of prediction and thus presents an element of uncertainty for the prediction model. When navigating confined spaces, such as docking in a harbor, the preferred control mode is still manual, although autonomous docking has been achieved for ferries[1]. Identifying situations where motion prediction could be feasible is therefore required and helps to limit the scope of the work shown in Fig. 1.1.

- **RQ5: Which operational situations could benefit from motion prediction?**

If the vessel is automatically controlled, the controller may be iterated along with predicted motion states and the corresponding reference trajectory, such as to inform the predictive model about future actuation signals (analogous to Model Predictive Control (MPC), but with a ML model). This has not been investigated in this thesis.

Data-based modelling has potential applications in neighbouring fields of interest for ship maneuvering, prediction and control. As such, the task of allocating forces requested by a motion controller may benefit from data-based modelling in terms of execution time. The allocation task is prevalent in all ship operations that involve low speed maneuvering and stationkeeping, which leads to the final research question.

- **RQ6: How can data-based modelling be used to facilitate constrained control allocation with limited a priori information?**

1.5 Scope of research

Motion prediction in itself is not tied to any specific application. As long as a model describing the propagation of motion due to the environment and/or state of the object is present, one may attempt to predict the subsequent motion. However, depending on the vessel operation mode, motion prediction may not be important. This generally includes navigating in open waters. It is therefore of interest to identify situations where motion prediction may have a positive impact on SA. This leads to the first objective, which is related to RQ5.

- **RO1: Identify ship operations that may benefit from predicting the near-future trajectory.**

The main objective of this thesis is to develop modelling methods that enhance trajectory prediction for maritime vessels, with the aim of adding to the SA of ship operators. RQ1-4 supports this line of inquiry and enables answering RO2.

[1]Autonomous ferry trials of Rolls-Royce and Wärtsilä, https://www.maritime-executive.com/article/rolls-royce-and-wartsila-in-close-race-with-autonomous-ferries, Date accessed 20-May-2021

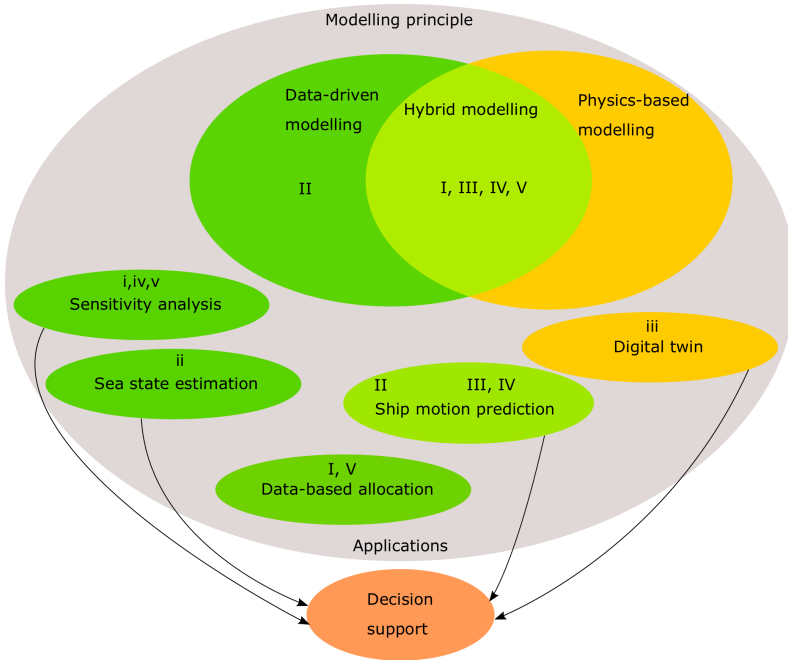


Figure 1.1: Overview of research items. The capital roman letters indicate publications on which the thesis is based, while lower case roman letters indicate associated publications.

- **RO2: Propose a combination of established dynamic motion models and data-based models that enhances motion prediction for ships.**

The secondary objective of this thesis is to support the use of ML in control-related tasks. To this end a control allocation scheme that allows for optimization and constraint handling at a fixed execution time is sought. This ties RQ6 to the final research objective.

- **RO3: Facilitate constrained control allocation using data-based modelling.**

Table 1.1 connects the ROs with the publications included in this thesis.

Table 1.1: Linking the research objectives to the published papers.

	I	**II**	**III**	**IV**	**V**
RO1		+	+	+	
RO2			+	+	
RO3	+				+

1.6 Thesis structure

The remainder of the thesis is structured as follows: Chapter 2 presents the platforms used for testing the models and algorithms, modelling regimes applied and the approach for control allocation using ML. Chapters 3, 4 and 5 present case studies that show the use of the developed approaches for ship motion prediction and control allocation and Chapter 6 presents conclusions and suggestions of future work. The first-author publications listed on page vii are shown in the appendices.

2 Methodology

2.1 Experimental platforms

For the duration of the Ph.D. project, three main experimental platforms have been used extensively to develop and validate the methods described in this chapter. Two simulator platforms have been used, which are described in Sections 2.1.1 and 2.1.2, as well as sampled data from the NTNU-owned R/V Gunnerus. Details regarding this ship is given in Section 2.1.3.

2.1.1 OSC simulator

The Offshore Simulator Centre[1] (OSC) provides simulator services for a variety of applications, including training for specific maritime operations and virtual prototyping. Although the commercial solution boasts a large array of features in terms of visualization and multi-station interfaces (crane operators, vessel operators, able-bodies, Remotely Operated Vehicle (ROV) operators etc.), the numerical models of the vessel, the simulated environmental disturbances and thrust-producing devices were utilized in this project. The simulator framework is written in the Java programming language and injection of algorithms can be performed through a plugin that allows interaction with simulated objects. This solution was applied for papers I and II.

To evaluate the motions of vessels, the OSC simulator utilizes the FhSim[2] software platform created by SINTEF. In this way the simulator provides access to a variety of reliable vessel models. A model of a multi-purpose offshore vessel was chosen for the tests performed in the OSC simulator. Its main dimensions are given in Table 2.1.

Table 2.1: Physical parameters of the OSC-simulated vessel.

Parameter	Description	Value
Loa	Length overall	93.79 m
Lpp	Length between perpendiculars	82.70 m
Bm	Breadth middle	23.00 m
dm	Draught	7.50 m
DWT	Deadweight	4925 t

[1]The Offshore Simulator Centre, https://osc.no/, Date accessed 23-May-2021
[2]The FhSim software platform, https://fhsim.no/, Date accessed 23-May-2021

2.1.2 OSP simulator

An open-source simulation platform[3] that provides co-simulation functionality was developed between major actors in the Norwegian maritime industry: DNV, Kongsberg Maritime, SINTEF, NTNU. It is built on the Functional Mock-up Unit (FMU) standard, which enables re-use of existing simulation models, in this case models of the ship- and thruster dynamics, and enables the user to easily inject custom code via FMUs into the simulation. The core output of the OSP joint industry project was the co-simulation orchestration library named *libcosim*, of which the Java wrapper *cosim4j* was used in this project.

The modularity of the FMU approach means that the user is free to select appropriate software tools for developing custom code and adopt existing simulation models. A ship dynamic model of the R/V Gunnerus, developed in the SimVal project [34], along with azimuth thruster models developed by Rolls-Royce Marine (now Kongsberg Maritime) in the ViProma project [35] made up the core of the simulation. FMUs providing motion control, control allocation and motion prediction were developed by the author to accommodate the case studies described in Chapters 4 and 5.

Co-simulation configuration (setting initial values, connecting inputs and outputs of FMUs) is performed using one of two standards: the OSP system structure or the System Structure and Parameterization[4] (SSP) standard.

2.1.3 R/V Gunnerus

The NTNU owns and operates a research vessel; the R/V Gunnerus[5]. A view from its starboard side is shown in Fig. 2.1. It is mostly used for research and educational purposes, throughout the NTNU system. Table 2.2 holds the physical dimensions of the ship. The ship was put into service in 2006 and in 2019 it was elongated by 5 m. As a result of the elongation, two versions of the vessel has been applied in this project. Both in terms of the numerical model and the sampled data. The dimensions given Table 2.2 refer to the elongated version, while Fig. 2.2 shows the original model.

Table 2.2: Physical parameters of the R/V Gunnerus.

Parameter	Description	Value
Loa	Length overall	36.25 m
Lpp	Length between perpendiculars	33.90 m
Bm	Breadth middle	9.60 m
dm	Draught	2.70 m
DWT	Deadweight	72 t

The vessel is equipped with three 450 kW generators that produce electric power for the three thrusters. Two main azimuth thrusters are located at the stern and a tunnel thruster at the bow according to Fig. 2.2. A dynamic positioning (DP) system from Kongsberg coordinates these thrusters during stationkeeping and low-speed

[3]The open simulation platform, https://opensimulationplatform.com/, Date accessed 23-May-2021

[4]Standard for configuring simulations that consist of FMUs, https://ssp-standard.org/, Date accessed 25-May-2021

[5]The R/V Gunnerus, https://www.ntnu.edu/oceans/gunnerus, Date accessed 23-May-2021

Figure 2.1: Starboard view of the RV Gunnerus.

maneuvering.

In this project the use of the ship has been limited to sampling data from specific operations. In paper IV data from a dedicated test was used, while paper III applied a data mining approach on all the data available from a year's worth of operations to extract sections of data preceding docking maneuvers. Note that all tests of algorithms on data originating from the R/V Gunnerus have been conducted offline, i.e. on data that is sampled some time ago.

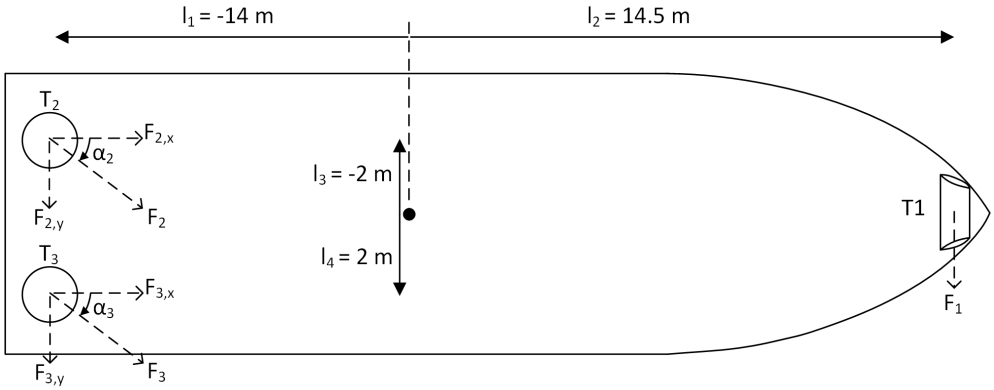


Figure 2.2: Thruster layout of the R/V Gunnerus.

As the vessel has a significant number of systems installed like thrusters, power generators, navigational sensors and a motion control system, the number of channels available for logging is high. In this project data originating from GPS receivers, a motion reference unit, compasses, a wind sensor and sensors measuring orientation and

rotational speed of thrusters has been utilized as input to the predictive models. Table 2.3 shows the various sensor channels.

Table 2.3: A subset of the data channels sampled onboard the R/V Gunnerus. The bottom two rows reflect the variables used in the data extraction procedure of paper III.

Data channel	**Unit**
North position	m
East position	m
Heading angle	deg
Surge velocity	knots
Sway velocity	knots
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
Wind direction	deg
Wind speed	knots
Course angle	deg
Speed over ground	knots
Port thruster rotational speed	RPM
Port thruster angle	deg
Starboard thruster rotational speed	RPM
Starboard thruster angle	deg
Tunnel thruster rotational speed	%
drive running	boolean
motor at zero speed	boolean

2.2 Modelling regimes

2.2.1 Physics-based models

A model describing the behaviour of an object is often based on existing, also termed a priori, information about the object. A structure of the model may be defined in terms of equations based on laws of physics. When the model structure is defined, proper parameters needs to be identified to reflect the specific object's dynamics. This process is referred to as model identification and takes place prior to applying the model for any application. Adaptive models may adjust its parameters with time if the object undergoes significant changes (environment, physical properties etc.). In this section the model of the ship motion will be elaborated.

This thesis is concerned with low and medium speed (0-3 m/s) trajectory prediction. A maneuvering model describing the motion of the ship in the Earth-tangential North-East-Down (NED) frame is therefore sufficient. It typically has a range of validity up to Froude numbers of 0.4. Applying the notation in [4] yields the vector containing the

ship north and east positions and the heading angle, $\boldsymbol{\eta}$. Velocities are given as $\boldsymbol{\nu}$ in the ship's coordinate frame. The components of this frame are the surge (longitudinal axis), sway (lateral axis) and yaw (rotation about the up-down axis) axes. As the two variables $\boldsymbol{\eta}$ and $\boldsymbol{\nu}$ are described in separate coordinate frames, the rotation matrix in (2.2) is needed to relate them.

$$\dot{\boldsymbol{\eta}} = \boldsymbol{R}(\psi)\boldsymbol{\nu} \tag{2.1}$$

$$\boldsymbol{R}(\psi) = \begin{bmatrix} \cos(\psi) & -\sin(\psi) & 0 \\ \sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.2}$$

The ship kinetic model in (2.3) describes forces (force along surge and sway axes and moment about the yaw axis) due to wind, waves, thrusters, hull friction, and inertia in the ship's coordinate frame.

$$\boldsymbol{M}_{RB}\dot{\boldsymbol{\nu}} + \boldsymbol{C}_{RB}(\boldsymbol{\nu})\boldsymbol{\nu} + \boldsymbol{M}_A\dot{\boldsymbol{\nu}}_r + \boldsymbol{C}_A(\boldsymbol{\nu}_r)\boldsymbol{\nu}_r + \boldsymbol{D}\boldsymbol{\nu}_r + \boldsymbol{D}_n(\boldsymbol{\nu}_r)\boldsymbol{\nu}_r = \boldsymbol{\tau}_c + \boldsymbol{\tau}_{wi} + \boldsymbol{\tau}_{wa} \tag{2.3}$$

Here, $\boldsymbol{\nu}_r = \boldsymbol{\nu} - \boldsymbol{\nu}_c$ is the velocity of the ship through water, and $\boldsymbol{\nu}_c = [u_c, v_c, 0]^T$ is the ocean current velocity. The rigid-body and added mass matrices are represented by $\boldsymbol{M}_{RB}$ and $\boldsymbol{M}_A$, Coriolis/centripetal forces are conveyed through the matrices $\boldsymbol{C}_A$ and $\boldsymbol{C}_{RB}$ and the linear and nonlinear damping matrices are $\boldsymbol{D}$ and $\boldsymbol{D}_n(\boldsymbol{\nu}_r)$, respectively. External force components are denoted by $\boldsymbol{\tau}$ and generally comes from the below three sources. Note that the force induced by ocean currents is included with the use of the relative velocity $\boldsymbol{\nu}_r$ in (2.3).

$\boldsymbol{\tau}_c$: The force produced jointly by the thrusters.

$\boldsymbol{\tau}_{wi}$: Force induced by wind acting at a certain angle relative to the ship.

$\boldsymbol{\tau}_{wa}$: Drift force induced by waves acting on the hull. This force is not measurable, but may be estimated using observers for certain operations such as DP.

The generalized control force, $\boldsymbol{\tau}_c$, is obtained by translating the individual thruster forces, that act at specific locations on the hull, to the ship's coordinate origin. Equation (2.4) shows the use of the thruster configuration matrix to perform the translation. Location and angle variables contained in the thruster configuration matrix are visualized in Fig. 2.2.

$$\begin{aligned} \boldsymbol{\tau} &= \begin{bmatrix} 0 & c(\alpha_2) & c(\alpha_3) \\ 1 & s(\alpha_2) & s(\alpha_3) \\ l_2 & l_1 s(\alpha_2) & l_1 s(\alpha_3) \\ & -l_3 c(\alpha_2) & -l_4 c(\alpha_3) \end{bmatrix} \cdot \begin{bmatrix} F_1 \\ F_2 \\ F_3 \end{bmatrix} \\ &= \boldsymbol{B}(\boldsymbol{\alpha})\boldsymbol{u}_f \end{aligned} \tag{2.4}$$

Models of the individual thruster forces vary in sophistication. Basic models apply a linear relationship between the control input u and the force output F [4]. Here, the control input is typically RPM, but may also be blade pitch angle or a combination of the two. Advanced models (referred to as 4-quadrant models [36]) account for factors such as RPM, time-varying advance velocity and propeller parameters such as diameter and geometry resulting in a closer approximation of the actual force produced by a propulsor. This modelling scheme was applied for the cases involving the R/V Gunnerus. In the OSC simulator a nominal relationship between RPM and propulsor force was interpolated to obtain thruster forces.

Wind force modelling exploits the above-water geometry of the vessel, through area-based coefficients, to determine the force induced on the vessel due to wind. Physical dimensions of the ship, force coefficients determined from e.g. wind tunnel tests and wind speed/attack angle contribute according to (2.5),

$$\boldsymbol{\tau}_{wi} = \frac{1}{2}\rho_a V_{rw}^2 \begin{bmatrix} C_X(\gamma_{rw})A_{Fw} \\ C_Y(\gamma_{rw})A_{Lw} \\ C_N(\gamma_{rw})A_{Lw}L_{oa} \end{bmatrix} \tag{2.5}$$

The projected area along the surge and sway axes are A_{Fw} and A_{Lw}, respectively. Force coefficients are given as C_X, C_Y and C_N and depend on the wind angle relative to the bow of the ship, γ_{rw}. V_{rw} is the relative wind speed and ρ_a is the density of air.

2.2.2 Data-based models

If existing information of the dynamics of a process is scarce or inaccurate, a model may be adapted purely based on the relations between certain input variables and target variables. The latter being the variables for which a numerical value, given input variables, is sought. Unlike the models described in Section 2.2.1 there are no fixed set of input parameters. This leads to the sub-domain of input/feature selection which aims to only extract features that have a substantial impact on the target variables. Limiting the input dimension is beneficial for both model complexity (number of parameters needed to produce a sufficiently accurate model) and input data dimension. The more input features, the greater the need for number of training data points.

The lack of explicit functional relationships between the input features and target values leads to the main challenge of data-based models; interpretability. This prompts questions regarding model parameter adaptation, structure and robustness.

- How will the model respond to edge cases?
- Have enough training data been included? And is the data representative for the environment in which the model will operate?
- Which hyperparameters lead to a well-performing model?
- Which features are important for a model to perform well?

These questions are not straightforward to answer. The open-ended nature of these questions also preclude any rigorous proof of convergence of the data-based model toward the true model of the ship. The designer of the data-based model is forced to

source data of sufficient quality and amount and apply optimization routines to select appropriate model structure, parameters and input features. By quantifying the response of the model through tests on representative datasets, metrics may contribute to our understanding of how the model will perform.

This thesis applies NNs as a ML modelling tool. The choice of using NNs is based on the widespread use within the ship motion prediction domain as well as in general data-based research. An array of well-documented platforms with active communities supports this choice. As a result of this, supporting frameworks exist that help answer some of the above questions. In the field of feature selection, methods exist to select subsets of features that contribute to the prediction performance of an ML model [37]. Model selection has been improved through various search strategies for hyperparameters [38], although model selection based on experience and trial and error is still a common approach.

Two of the most commonly used network types for regression have been applied: the Multi-Layered Perceptron (MLP) [39] and the recurrent Long Short-Term Memory (LSTM) network [40].

MLP

The MLP network is composed of an input layer, one or more hidden layers and an output layer. Fig. 2.3 depicts a simple instance of such a network. Each hidden layer contains one or more nodes that applies an activation function to the sum of the inputs from the previous layer. If the MLP is fully connected, which is the common way of implementing it, connections from every node at the preceding layer is made to the successive layer. Each node in the hidden layer(s) also has an associated bias value that shifts the origin of the activation function. The sigmoid, hyperbolic tangent and Rectified Linear Unit (ReLU) activation functions are popular choices for the hidden layer nodes. With the ReLU being a fairly recent addition which lends itself to training efficiency in networks containing several layers, also known as Deep NNs (DNNs).

$$o_k = a(\boldsymbol{w}_{ik}\boldsymbol{x}_i + \boldsymbol{w}_{jk}\boldsymbol{x}_j + \boldsymbol{b}_k) \tag{2.6}$$

In relation to Fig. 2.3 the output of node k, o_k, is calculated according to (2.6). The mapping $a()$ represents the activation function, $\boldsymbol{w}$ is the weight vector, $\boldsymbol{b}$ is the bias vector and $\boldsymbol{x}$ is the input signal.

Learning in an MLP network, and indeed any network that applies supervised learning, is performed through multiple passes of a dataset that contains input features and associated target values. The target values are the desired output of the network at the output layer. An error signal is generated based on a loss function, typically mean squared error (MSE), which accepts the desired output of the network and the network prediction. This error signal is then propagated backwards through the network and appropriate corrections to the parameters of the network (weights and biases) are performed based on the gradient of the error signal with respect to the parameters.

An overarching goal of data-based models is to be able to represent the system it models on new data, ie. data that it has not seen during training, accurately. This is known as the generalization ability of the model. Several techniques exist to approach this challenge. Avoiding overfitting the model on the training data can be managed

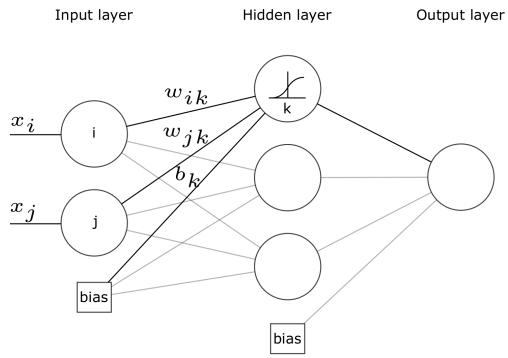


Figure 2.3: Structure of a general MLP. The figure illustrates the components that a general MLP consists of. Typically, deeper (more hidden layers) and wider (more nodes per hidden layer) networks are used.

through:

- stopping the training procedure based on the relative improvement of the model's performance on some validation dataset
- managing the complexity of the model (number of parameters)
- introducing regularization (penalizing weight magnitude)
- feature selection (including only relevant input features)

LSTM

RNNs are well suited for modelling systems that are sequential in nature [41]. This is due to their ability to maintain states within a network due to feedback/recurrent connections. Earlier versions of RNNs suffered from stability and training deficiencies due to the vanishing gradient problem [42]. This problem was addressed by Hochreiter and Schmidhuber [40] where gates managing the flow of error was introduced. The resulting network was named LSTM and has since been the state-of-the art within modelling of sequential processes using NNs [43]. What makes the LSTM unique relative to standard RNNs is the use of gates in each LSTM block that manages the signal flow (see Fig. 2.4). Specifically, there are three main functions within the LSTM: forgetting (orange dashed sector), adding new information to the internal state (purple dashed sector) and determining the level of the state to output (green dashed sector).

2.2.3 Hybrid models

As a reaction to the disparate properties of the models described in Sections 2.2.1 and 2.2.2, in terms of white/black-box modelling and fidelity issues, some researchers have turned to hybrids that comprise aspects of both domains. Model fidelity issues may result from applying a simplified physics model or a sub-optimal training scheme configuration for a data-based model. This was addressed by Psichogios et al. who applied an NN model to feed a first-principles-based model with an estimate of a process state that

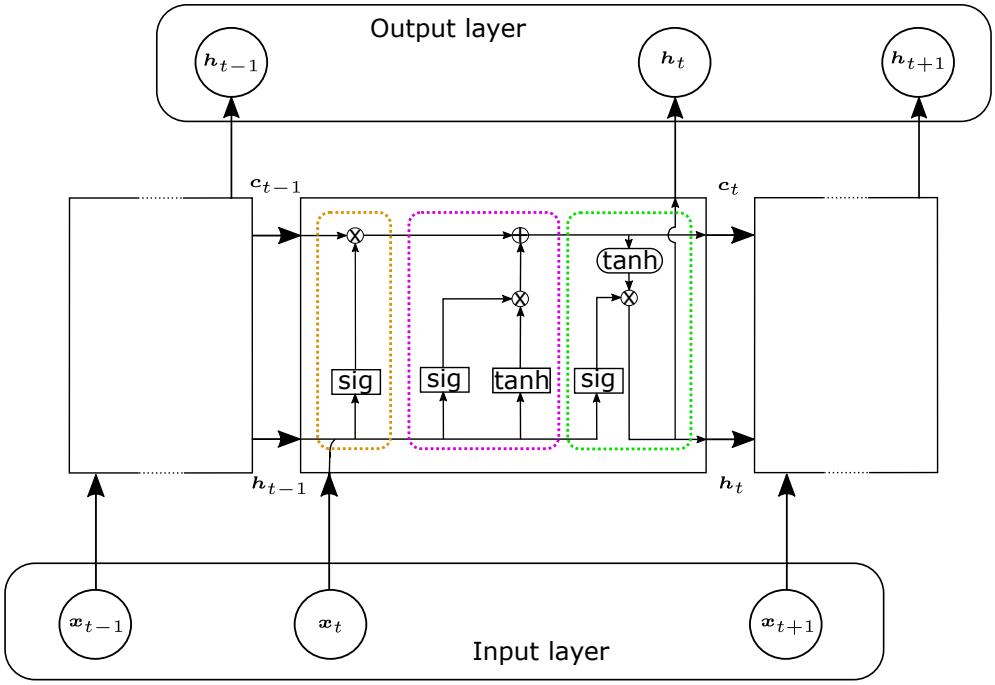
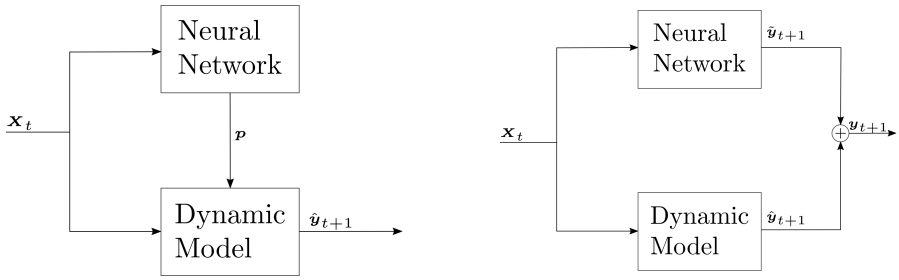


Figure 2.4: Structure of LSTM node.



(a) Model/Process Parameter [7, 8]

(b) Parallel [11, 12]

Figure 2.5: Two ways of combining physics-based models and ML models as provided by relevant literature. $\boldsymbol{X}_t$ are input features, which may be unique for each model class. $\boldsymbol{p}$ are dynamic model parameters or a partial process state, while $\hat{\boldsymbol{y}}_{t+1}$ is the prediction made by the dynamic model predictor. $\tilde{\boldsymbol{y}}_{t+1}$ is the NN residual prediction, which added to the dynamic model prediction makes up the parallel hybrid prediction, $\boldsymbol{y}_{t+1}$.

does not easily lend itself to physics-based modelling [7]. This type of hybrid structure is shown in Fig. 2.5a and referred to here as the parametric hybrid approach. It is parametric in terms of a process state of a physics-based model (as in [7]) or in terms of model parameters. The latter option was applied by van de Ven et al. in the maritime domain to estimate the damping matrix of an underwater vessel [8].

Raissi et al. [44] applied a different approach, where the governing equations entered the network training procedure as loss functions. In other words, the physics-based model does not contribute directly to the outputs of the hybrid model. Here, an automatic differentiation scheme enabled the determination of derivatives of the output states of the NN. The derivatives were required as inputs to the physics-based equations that contribute as part of the total loss function. This approach is generally described by (2.7) which indicates how a physics-guided/informed NN could be obtained through the use of governing equations in the network's loss function [9].

$$Loss = Loss_{TRN}(\boldsymbol{y}_{t+1}, \hat{\boldsymbol{y}}_{t+1}) + \lambda R(\boldsymbol{W}) + \gamma Loss_{PHY}(\hat{\boldsymbol{y}}_{t+1}) \tag{2.7}$$

$Loss_{TRN}$ represents the standard supervised training loss of a network (MSE or similar metrics), the mapping $R()$ is a measure of the model complexity and λ weights the emphasis on reducing the complexity. Physical consistency is encouraged through the third term, $Loss_{PHY}$, which is weighted by γ relative to the first and second term of (2.7). Thus the network training objective of minimizing the overall loss, $Loss$, incorporates knowledge of the physics of the process.

The hybrid modelling approach used in this project aims to compensate the error made by a kinetic model of a surface vessel. This is visualized in Fig. 2.5b. An extensive survey of various methods of integrating physics-based and ML-based methods is given in [45].

2.3 Ship motion prediction approaches

Papers II,III and IV revolve around time-series prediction of the trajectory of ships. Therefore, this section will outline the applied kinetic, data-based and hybrid predictors.

2.3.1 Kinetic model predictor

An often-used kinetic model of a ship is described in Section 2.2.1. It establishes the relation between forces induced by disturbances, actuators and intrinsic model parameters and the motion (velocity and acceleration) of the ship. Once the numerical values of the model parameters have been identified, propagating the position by numerical integration of the acceleration may be performed [46]. In such a way, predictions of future motion states may be obtained. Fig. 2.6 illustrates this approach. Several elements contribute to its accuracy: the disturbances can not be fully measured, the model fidelity, the identification procedure uncertainty, accuracy of the integration scheme etc. However, the end result is a deterministic model that, given proper input, can output predictions in an efficient manner. A caveat of this predictive approach is that forces from the actuators are unknown for future time instances. Thus, scenarios where the future commands are known (see Section 4) are required, or the user must accept that the predictions are valid for the present command vector.

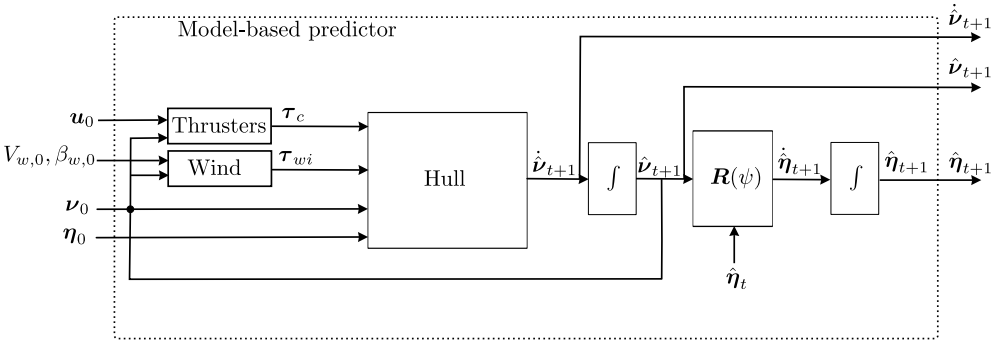


Figure 2.6: Generating trajectory predictions using the kinetic model.

2.3.2 Data-based predictor

Establishing a ML model for predicting the states of a ship requires the completion of a few basic steps prior to performing training. Compared to the kinetic model, no fixed set of sensor values are applied. The parameters of the model are also not specified.

Data preparation and feature selection

Perhaps the most critical task is to accumulate data that holds information relevant for determining the future states. In this thesis, the goal of the predictors are trajectory prediction. This entails predicting the future position and possibly heading. In navigation systems, position is often given as an absolute value in terms of Latitude and Longitude. However, in a ML setting this representation is not ideal. Vessels usually travel long distances, leading to a large input range for position. A better representation

would be to transform the global coordinates into local coordinates, relative to eg. the earth-tangential, NED coordinate frame. This leads to a much narrower input range in terms of position.

If heading is considered as an input or target variable the discontinuity at 360/0 degrees must be handled. The same applies to wind direction and other angle-based measurements. A transformation of some kind is in order if any of the states cross the discontinuity. An example of transforming the wind measurement is given in (2.8).

$$\boldsymbol{V}_w = \begin{bmatrix} \cos(\gamma_{rw}) \\ \sin(\gamma_{rw}) \end{bmatrix} V_{rw} \tag{2.8}$$

Here, V_w holds the longitudinal and lateral relative wind speed. As part of the basic operations performed on the raw data, signal processing techniques such as outlier detection and scaling are appropriate next steps. This results in data that have similar magnitudes which is a preferable trait of data used for ML.

A set of informative features should make up the input variables of any NN. It is generally beneficial to minimize the number of features since it facilitates better generalization, increases interpretability and reduces complexity of the NN model [37]. Feature selection can thereby be used in a semi-automated way (a method ranks features and the designer sets a threshold). Depending on the desired feature relevance or desired input dimension, a set of features gets selected. This approach was applied in paper II using mutual information as a means of ranking the raw features [47].

Model selection

ML models do not impose constraints on the structure of the model for a given modelling problem. Therefore, any model class (NN, support vector machine (SVM), linear regression) combined with any configuration of its parameters are possible representative candidates [48]. In this work NNs were considered to be a good model class due to their widespread use within research and well-developed tools with active user forums. In addition, the LSTM sub-class of NNs lend itself naturally to sequential data such as time-series prediction problem [49].

Model selection refers to the selection of model class as well as parameters that dictate its structure and behaviour. This process may concern every conceivable parameter that the user may change, known as hyperparameters, or more commonly, a subset of the hyperparameters. The hyperparameter space may be explored manually, or by some search method designed to reveal optimized parameter configurations [50]. This requires iterations of the model training procedure for distinct hyperparameter configurations, leading to a large computational cost. Common search methods are random search [51] (explores the search space efficiently), grid search (computationally expensive, but covers all combinations) and bayesian search [52] (makes informed decision on search direction, but requires some extra computation per model evaluation).

2.3.3 Hybrid predictor

Section 2.2.3 introduced a few modelling schemes for combining ML models with existing first-principles-based models. In this section the implementations of the schemes from papers III and IV will be presented.

What to represent with a ML model

Modelling only part of the physics-based model with NNs (see Fig. 2.5a), as suggested in [7, 8], is a logical approach. It assumes that the remaining part of the physics-based model is adequately identified. For a ship this can be exemplified by the relative uncertainty of damping parameters versus the mass distribution. Sources of disturbance must also be accessible, such that the uncertainty can be isolated at the partial NN model.

The experimental platform detailed in Section 2.1.3 has limitations in its sensor array that precludes making use of partial ML models or partial parameter updates. This is related to the unmeasured environmental impact of waves and ocean current. This leaves two options: either neglecting the effect of current and waves and attempt to model the most uncertain parts of the model (such as damping matrices, see Section 2.2.1), or lump these uncertainties together with the effect of current and waves (the parallel hybrid structure). The latter approach was taken in this project and the subsequent sections will elaborate this implementation. Within the research group of the author of this thesis, called the Intelligent Systems Laboratory at NTNU Ålesund, research on ship motion prediction is a key area of interest. Thus, in parallel to paper IV, Wang et al. [10] presented a cascaded hybrid approach for trajectory prediction. Here, the output of the kinetic model predictions act as input to an NN.

Generating training data

To acquire data for training, the dynamic model predictor, shown in Fig. 2.5b, must be evaluated on the recorded data to produce its predictions, $\hat{\boldsymbol{y}}_{t+1}$. These predictions are then used to obtain the target variables used for supervised training of the parallel hybrid predictors. Fig. 2.5b shows the signal flow during inference.

Iterative or direct prediction

The dynamic model presented in Section 2.2.1 outputs the acceleration of the vessel in three dimensions. When this model is applied to get the future motion of the ship, a numerical integration scheme may be applied. Typically, this integration scheme outputs the estimated/predicted acceleration state one sample interval ahead. An integration scheme would reiterate the equations several times to achieve this.

For the NN predictor, two approaches exist: either output predictions at one step ahead, termed single-step prediction, or output predictions multiple steps ahead at once, termed multi-step prediction. The first approach is more intuitive, but is prone to accumulating prediction errors as it requires an AR structure (ie. making use of predictions from the previous sample time in the input vector). Making multiple predictions from an initial time instance does not have this deficiency, but leads to a more complex model (the target value vector increases in dimension and the NN must therefore adapt to more complex relations within the input/output variables). The two approaches were applied in paper III and IV, respectively.

2.4 Control allocation

In Section 1.3 existing control allocation methods were outlined. This section describes the proposed NN allocator. The key enabling features are the various loss functions that

contribute to shape the weights of the network.

The primary function of the control allocation module is to make the thrusters jointly produce the requested generalized force [26]. Secondary objectives include limiting these forces to adhere to the inherent limitations of each actuator. Depending on the actuator type, this may include maximum/minimum force and angle and their respective rates. Generating force results in increased fuel consumption and emissions. Weighting the use of energy against the positioning/maneuvering performance is an important trade-off.

2.4.1 Model structure

The structure of the NN allocator proposed in this Ph.D. project was inspired by the work in [53]. The general structure of the allocator is an Autoencoder that receives force commands from a motion controller and attempts to reproduce this force at its output nodes. Fig. 2.7 shows the block diagram of such a network. In the process of reproducing the forces, $\boldsymbol{\tau}$, the loss functions intervene to shape the encoder predictions as well as the decoder predictions. In the context of autoencoders the encoder output is known as the code or the latent space.

The layers between the input data and latent space are referred to as the encoder. They transform the input data into latent variables. The transformation is reversed in the decoder to reconstruct the original input data. Normally this structure is used to compress data, but here the latent dimension is greater than the input dimension due to the number of actuator control signals. Also, the natural division of the network into two parts gives the training procedure freedom in terms of shaping the output of the encoder. During inference/allocation the output of the encoder outputs thruster commands given force requests from the motion controller. Had a standard NN been applied to map generalized force (the input vector) to actuator commands (output), there would be less freedom in constraining the actuator commands in a supervised training setting. In other words, the training data would have to reflect the desired constraints of the allocator.

2.4.2 Data generation

An important consideration in modelling such a system is the procurement of data. The input space comprises the three components of the generalized force, which have a large force range. Manually exploring the entire generalized force space (at a reasonable resolution) using a simulator would take a very long time. Doing this on a real vessel would be next to impossible. Procuring data from real DP operations would lead to transferring the behaviour of the allocator implemented on the ship. This would also require having a functioning allocator (chicken and egg problem). The benefit of the mentioned sampling approaches is that they inherently capture the dynamics of the actuators on the ship.

Based on the above considerations a synthetic dataset was generated, which does not include the actuator dynamics. However, loss functions constraining the rates and magnitude of the allocated commands were considered to mitigate this problem. A random uniform sampling strategy was selected to obtain samples that cover an entire sub-space of the allowable thruster commands. Converting these commands using the thruster configuration matrix in (2.4) resulted in actionable force requests. They were used as inputs and targets for the supervised learning scheme.

2.4.3 Constraints as loss functions

NNs adapt internal weights, which in turn shape its response, during the training phase. Loss functions are a means of quantifying the performance of a model and the weights of the network are adapted to minimize the loss function. Several loss functions for the NN control allocator were specified to cover the range of constraints imposed on the allocator. These are briefly introduced in this section. A more thorough explanation is given in paper V.

Loss functions L_2-L_5 penalize excursions beyond the constraints supplied by the user. These are listed below and their location in the network is given in Fig. 2.7.

L_2: Penalizes all force/azimuth angle excursions beyond a fixed limit during training. The loss is given as the difference between the absolute value of the predicted command and the maximum command.

L_3: Rates exceeding the inherent limit for each thruster are penalized according to the difference between predicted rate and maximum rate.

L_4: This constraint acts on the forces of each thruster raised to the power of 1.5. It is active in the entire force range.

L_5: Wakes from adjacent thrusters may hamper the ability of a thruster to produce force [54]. To avoid this, one may impose constraints in terms of allowable azimuth angles. Here, this loss is binary (value of 1 if constraint is violated). Loss functions that impart graded loss based on the magnitude of constraint violation may also be considered.

The remaining loss functions, L_0 and L_1, are functions of the requested generalized force. The former contribute to the overall minimization goal of the Autoencoder. The latter applies the thruster configuration matrix to transform predicted commands, $\hat{\boldsymbol{u}}$, into resulting generalized force. It therefore acts to minimize the error of the predicted allocated generalized force relative to the requested generalized force, ie. the allocator's primary goal.

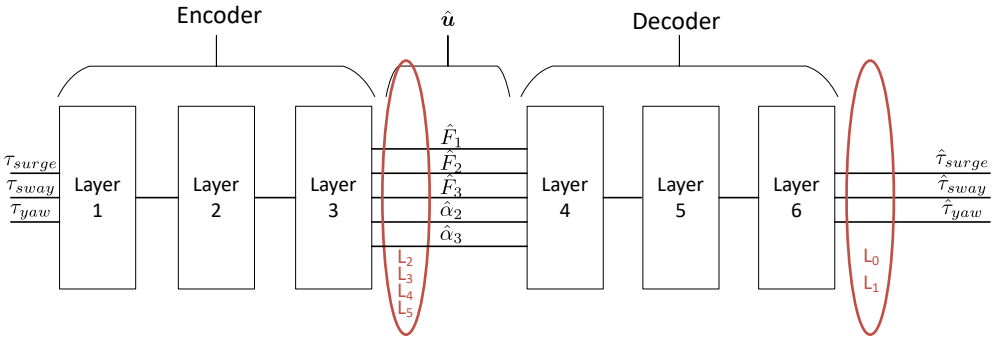


Figure 2.7: The structure of the Autoencoder used for control allocation. The red ellipses indicate the variables that go into loss functions (marked in red text).

3 Case study: Dead reckoning

Dead reckoning (DR) is a procedure that allows the position of the object, in this case a ship, to be continuously estimated during an outage of the sensory inputs of the position. In this way it offers the ability to bypass the standard way of obtaining the position of the ship for a short period of time. This research constitutes the initial study of using data-driven models for position prediction and the paper may be found in Section II of the Appendix.

GNSS consists of a constellation of satellites that emits signals which enable GNSS receivers to determine its position[1]. If a vessel provides redundant systems for obtaining measurements of the position, DR is less relevant. This is due to having redundancy in terms of sensors, which leaves a total outage less likely. Such auxiliary position references could be based on relative position sensors such as seabed hydroacoustic transponders or reflectors located on floating or fixed structures. Experiments related to this data-driven DR method was conducted in the OSC simulator described in Section 2.1.1.

Similar to the scenario presented in Section 4.1.2, a failure during stationkeeping occurs. In this case the failure is related to external sensory inputs, leaving the maneuvering ability of the vessel and a subset of the motion sensors intact. As such the case study presented in this chapter relates to RO1, but the trajectory predictions are limited to one-step predictions for maintaining estimates of the ship position.

3.1 Data-driven DR

Even though the external position measurements, obtained through GNSS signal processing, are lost, internal sensors remain active. In the proposed data-driven DR method the ship's heading, wind speed/direction and propulsion system signals may still be sampled. Thus, for each step of the data-driven DR algorithm, the longitudinal and lateral velocities of the ship are predicted. These are given as $\hat{\boldsymbol{v}} = [\hat{v}_{lon}, \hat{v}_{lat}]$ in Fig. 3.1.

The predicted velocities are rotated by the heading angle, ψ, to obtain the velocities relative to the north and east axis of the NED frame and added to the previous predicted position, $\hat{\boldsymbol{p}}[k]$. k indicates the discrete sample time. Δt is the sample interval and may be set to reflect the sample interval of the original position measurements. Equation (3.1) shows the iterative propagation of position.

$$\hat{\boldsymbol{p}}[k+1] = \hat{\boldsymbol{p}}[k] + \boldsymbol{R}(\psi)\hat{\boldsymbol{v}}[k+1]\Delta t \tag{3.1}$$

Recurrent NN layers were chosen to model the relationship between the input vector

[1]The GNSS, https://www.euspa.europa.eu/european-space/eu-space-programme/what-gnss, Date accessed 15-June-2021

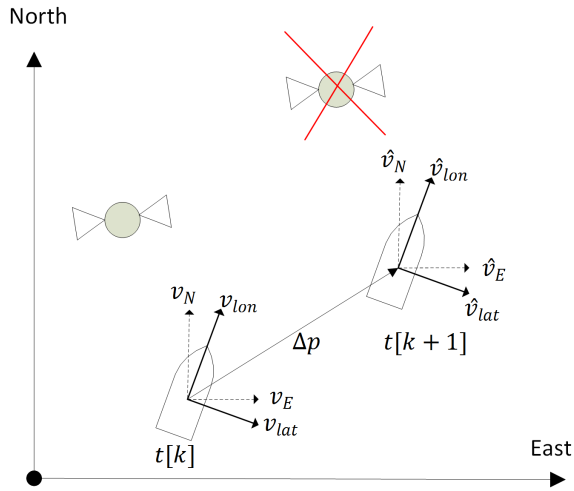


Figure 3.1: A switch from normal operation (t[k]) to loss of GNSS system, requiring a DR system to estimate the position at the next step without an absolute position measurement.

(wind speed/direction, heading angle, thruster measurements) and the output vector (longitudinal/lateral ship velocities). They were selected based on their inherent ability to represent sequential data and handle lags between a change in the input vector and the corresponding output change. In addition the prediction problem was partitioned into two NNs: one for each of the components of the output $\hat{\boldsymbol{v}}$. This was done to allow for customized input vectors for each NN. This will be elaborated in the following section.

3.2 Data and pre-processing

A sequential simulation procedure was applied to explore the DR performance at varying degrees of wind and wave disturbances. Four distinct disturbance levels were applied according to Table 3.1.

Table 3.1: The parameters of the sea states simulated at each discrete weather direction.

Significant wave height (Hs)	Wind velocity
1 m	2 m/s
2 m	4 m/s
3 m	7 m/s
4 m	11 m/s

Each of the four disturbance levels were simulated for 14 minutes to minimize the effects of transients in the evaluation of the DR performance. This process was run for wave/wind angles in the range 0-360° at 30° intervals. In order to approach the performance of real sensors, additive noise was applied to the position, velocity and heading measurements. The effect of noise on position measurements are shown in Fig. 3.2.

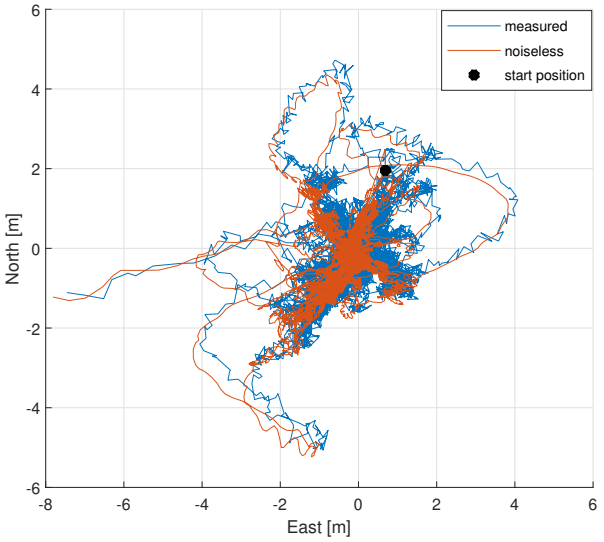


Figure 3.2: Visualization of the vessel position. The noiseless position measurement (red line) is only included to provide a reference to the raw GNSS position output (blue line).

After having obtained the data from the simulation procedure, a scaling procedure is required to ensure that each input feature and target have a similar range. The z-score normalization, also known as standardization, was applied for this purpose according to (3.2).

$$\boldsymbol{x}' = (\boldsymbol{x} - \bar{\boldsymbol{x}})/\mathrm{std}(\boldsymbol{x}) \tag{3.2}$$

Here, the standard deviations of each of the input features (calculated on the training data, $\boldsymbol{x}$) are represented by $\mathrm{std}(\boldsymbol{x})$ and the mean values are given as $\bar{\boldsymbol{x}}$. Similarly, the output variables $\hat{v}_{lon}$ and $\hat{v}_{lat}$ are standardized based on the metrics from the training data (see (3.3)).

$$\boldsymbol{y}' = (\boldsymbol{y} - \bar{\boldsymbol{y}})/\mathrm{std}(\boldsymbol{y}) \tag{3.3}$$

As a final step before searching for optimized model parameters, the input vectors of the two individual DR predictors are examined through Mutual Information (MI). It measures the dependence between two variables and thereby enables filtering out the less significant input features relative to each of the two target variables. Applying a user-defined threshold value, a certain percentage of the original input features are kept. In this case study a threshold value of 0.4 resulted in 4 features being selected for each NN in the DR approach. To model $\hat{v}_{lon}$, the inputs *wind angle*, *wind speed*, *main thruster power* and *main thruster RPM* was selected. And to model $\hat{v}_{lat}$, the inputs *wind angle*, *wind speed*, *bow thruster power* and *stern thruster power* was selected. Methods like the

MI may be applied to answer RQ3 although there is no one set of input features that is superior for a certain operation.

3.3 Model selection

A semi-automated procedure was selected to search for optimized configurations of learning rate and number of blocks in an LSTM single-layer network. Considering the number of potential search parameters, some intuition about which are most influential is required to limit the computational cost of this search. Admittedly, the parameter range is quite constrained, but it searches two key parameters according to [43]. Based on the performance of each of the evaluated searched model configurations, the two below configurations were applied when testing the proposed NN DR.

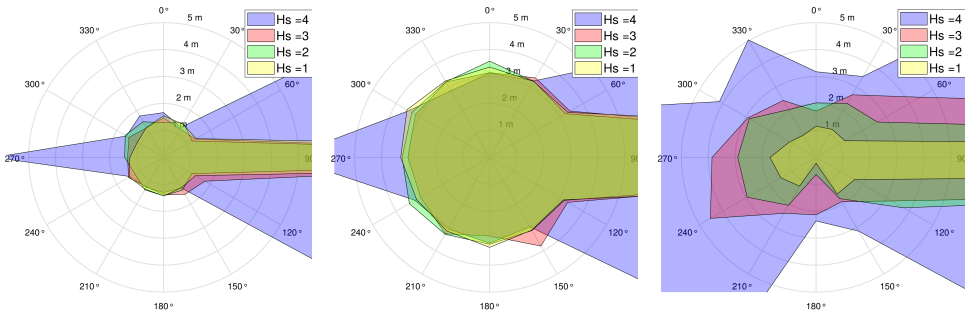
- surge velocity: block number = 43, learning rate = 0.0070
- sway velocity: block number = 26, learning rate = 0.0165.

3.4 Predict position during GNSS failure

In addition to the LSTM DR method described above, where LSTM blocks were applied in the hidden layer, two other model classes were tested. A single-layer feedforward network (SLFN) and a Kalman Filter (KF) incorporating mass and damping matrices of the linearized dynamic model for the vessel given in Table 2.1. The only difference between the SLFN DR and the LSTM DR was the layer type and therefore also hyperparameters.

Fig. 3.3 shows the results in terms of mean position error, calculated on a one-minute DR period, for the three methods using (3.4). The LSTM network provides the most consistent and accurate results in terms of position estimates for all directions and wave heights (Hs, given in meters). The exaggerated errors seen for wave heights of 4 m are attributed to the saturation of some of the thrusters leading to a significant position deviation.

$$\bar{e}_{dist}[k] = \frac{1}{N}\sum_{k=1}^{N}\sqrt{(\hat{p}_n[k] - p_n[k])^2 + (\hat{p}_e[k] - p_e[k])^2} \tag{3.4}$$



(a) LSTM (b) SLFN (c) KF

Figure 3.3: Results of mean position estimation error given in the horizontal plane for the LSTM, SLFN and a KF model for case study 2 in paper II. Each data point shows the mean position estimation error during a one-minute DR period. Hs denotes the significant wave height in meters.

4 Case study: Hybrid prediction

This chapter presents research results from papers III and IV. Together, they seek to display the benefit a hybrid approach to ship motion prediction and answer RO2. The data used for testing originates from the R/V Gunnerus presented in Section 2.1.3, although some initial tests were conducted in simulators. A description of how the data was acquired comes first, followed by an introduction of the methods and results obtained.

4.1 Data extraction

Two separate operation types were identified as operations where ship motion prediction could contribute to enhance the overall situational awareness: docking the ship and a power system failure. This relates to RO1. Docking a ship inevitably involves entering ports that are populated by other vessels as well as converging towards the fixed docking location. Predicting the future trajectory of the ship could enable both safer maneuvering and more efficient docking. In the case study described in Section 4.1.1, a prediction horizon of 30 seconds was considered sufficient time for the ship operator to make trajectory corrections. This is substantiated by Perera et al. for general low speed maneuvering [15]. In reality, the length of the prediction horizon should be based on the maneuverability and size of the vessel. When a ship is in the stationkeeping mode, it will remain within the vicinity of its desired position. Thus, trajectory prediction is not required. However, if a failure occurs, e.g. a power failure as described in Section 4.1.2, the ship will drift. Since stationkeeping is often connected to maneuvers close to fixed offshore installations or deployment of seabed installations, a drift-off could have severe consequences. A trajectory predictor for the power failure case could therefore be applied prior to initiating an operation to gauge the direction of drift.

4.1.1 Docking

The initial study of hybrid ship motion prediction revolved around docking operations. In this respect the R/V Gunnerus was the ideal experimental platform since it frequents several ports along the west coast of Norway continuously. Its data logging systems, and the availability of this data (spanning several years in the past), were major facilitators of this study. A bulk of the data spanning one year of operations was selected and from it, each docking operation was pinpointed by looking at a few indicative sensor channels (since no record of the docking time instance is kept). These sensor channels were the overall ship speed and the two boolean power system signals *drive_running* and *motor_at_zero_speed*. Docking was considered to be completed when the ship speed was below 0.1 m/s, and the two boolean signals were *False*.

From the original one-year history of data, 88 individual docking operations were

extracted. Their geographical distribution is shown in Fig. 4.1. A time window of 1000 seconds prior to the identified docking time instance was selected based on the typical maneuvering profile. In that period deceleration was initiated and low-speed maneuvering towards the dock commenced. A bias towards docking in the port of Trondheim, Norway, was observed since this is were the R/V Gunnerus receives most of its business.

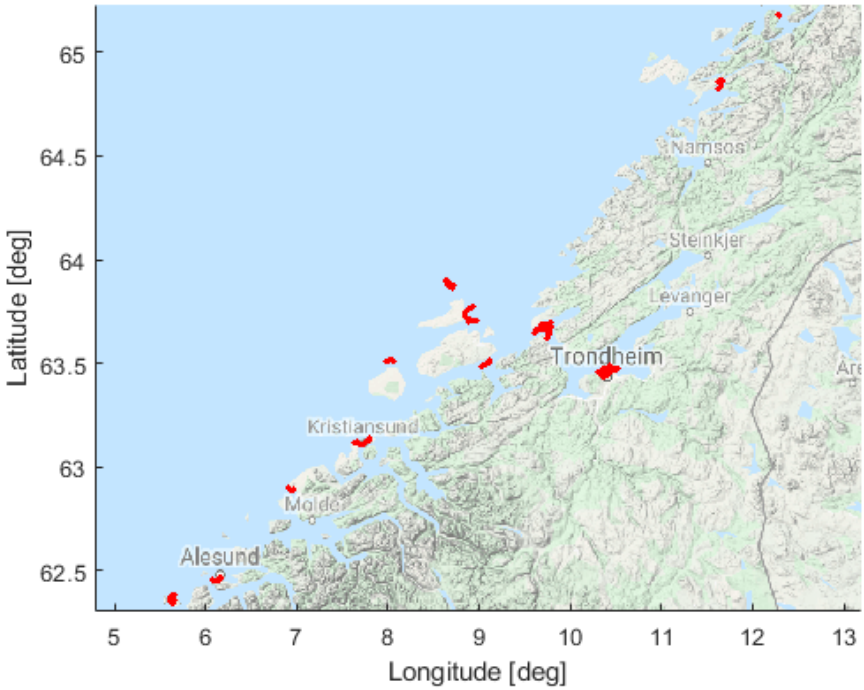


Figure 4.1: A visualization of the docking locations of the R/V Gunnerus throughout the one-year dataset.

4.1.2 DP power failure

Towards the end of 2019 the R/V Gunnerus was hired by the Department of Ocean Operations and Civil Engineering at NTNU Ålesund to conduct a series of tests. This to support research within data-driven modelling and analysis of ships. During this cruise a test was conducted to simulate a power failure during DP. Fig. 4.2 shows the path of the ship before and during this test.

Important considerations for such a test includes the effect of unmeasured effects on the ship, such as the influence of current and waves on its motion. This motivated the use of a relatively small amount of data sampled prior to the simulated failure. The effects of the local environmental disturbances could therefore be assumed to be similar

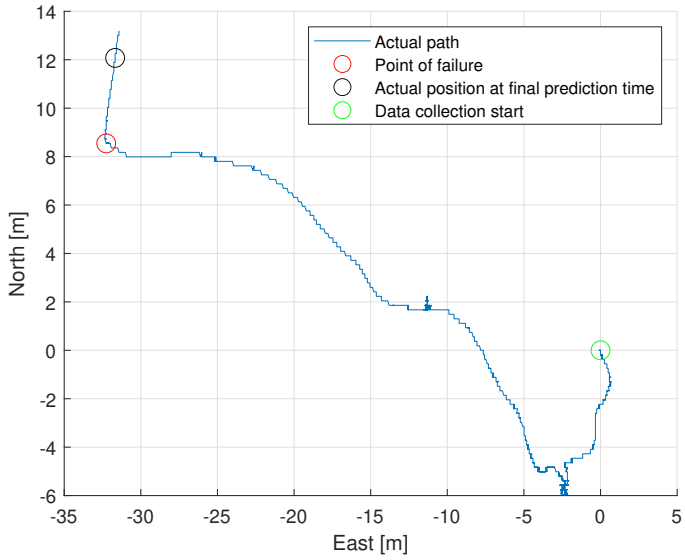


Figure 4.2: Positions sampled prior and during the DP power failure test.

throughout the test.

4.2 Experiments

Two different prediction schemes were applied in papers III and IV: the direct and iterative, respectively. Sections 4.2.1 and 4.2.2 outline the design of the hybrid predictors and present results.

4.2.1 Hybrid docking predictor

Isolating the origin of prediction errors is impossible if all the force-inducing effects on a vessel are not accounted for. In other words, applying the model/process parameter hybridization of Section 2.2.3 would result in lumping the effects of several phenomena (uncertain damping coefficients and wave/wind impact) into e.g. the parameters of an NN-derived damping matrix. The parallel hybridization inherently considers only the lumped prediction error, and thus presents a more intuitive way of modelling the future motion of a vessel. Alternatively, the loss function hybridization could be applied. This approach would, however, require some additional tuning to take into account the level of which to accommodate NN parameter adaptation based on the physical inconsistency of the NN output during training.

Input data

The selection of input features was not based on feature selection techniques for the NN docking predictor. Table 2.3 shows the 21 sensor channels that were applied. Due to having multiple repetitions of most of the docking locations in the training data,

the north/east position (relative to the docking) were included. Potentially, this could lead to the network learning maneuvering policies and thus achieving higher predictive performance when docking at familiar ports. The heading angle and surge/sway speed contain information that is correlated with the change in position. The pitch/roll angle, the heave (up/down) displacement and their respective rates may convey information about the wave state (although limited in ports) and also changes in loading conditions of the ship. Effects of wind were included through the measured wind angle and wind direction. Although the ocean current speed and direction could not be measured directly, information about its direction could be conveyed through the use of the course angle and heading angle for certain maneuvering conditions. Hence the inclusion of course angle in the input data. When predicting several seconds ahead (30 in this case), the future commands are not known to the predictor. This applies to both the kinetic model predictor and the NN predictor. This is a major obstacle of this approach, and a feature that must be communicated to the operator if it is to be applied on a real ship. For automatically controlled docking scenarios the future commands may be obtained through running the controller in parallel with the incrementing prediction horizon, similar to model predictive control schemes.

Model selection

A combination of the kinetic model in Section 2.2.1 and a multi-layered, ensemble LSTM was proposed as representative model for the docking trajectory predictor (see Fig. 4.3). Ensembles, a set of 3 LSTM NNs where the average prediction is output to correct the kinetic model predictions, were applied to get more stable predictions. Similar to the model selection procedure in Section 3.3 a hyperparameter search was conducted. The option of adding an additional hidden layer was included and resulted in the below search space:

- Learning rate: $[10^{-4}, 10^{-1}]$
- Hidden layers: [1, 2]
- Number of nodes: [10, 500]

Results

Having trained the NN predictor on the north/east prediction error output by the kinetic model predictor on 68 individual docking datasets, its performance was evaluated on 20 docking datasets. An evaluation metric was constructed to gauge the prediction error for increasing prediction horizons, $i \in [1, 30]$, according to (4.1). M is the number of samples and also the number of 30 second prediction instances. N and E indicate north and east positions.

$$\bar{y}_{err,i} = \left(\sum_{j=1}^{M} \sqrt{(N_{ij} - \hat{N}_{ij})^2 + (E_{ij} - \hat{E}_{ij})^2} \right) / M \tag{4.1}$$

The average prediction error for each of the 20 dockings is plotted against the prediction horizon length in Fig. 4.4. It also displays the average errors of a predictor that only applies the kinetic model (dashed red line) and the hybrid predictor (dashed

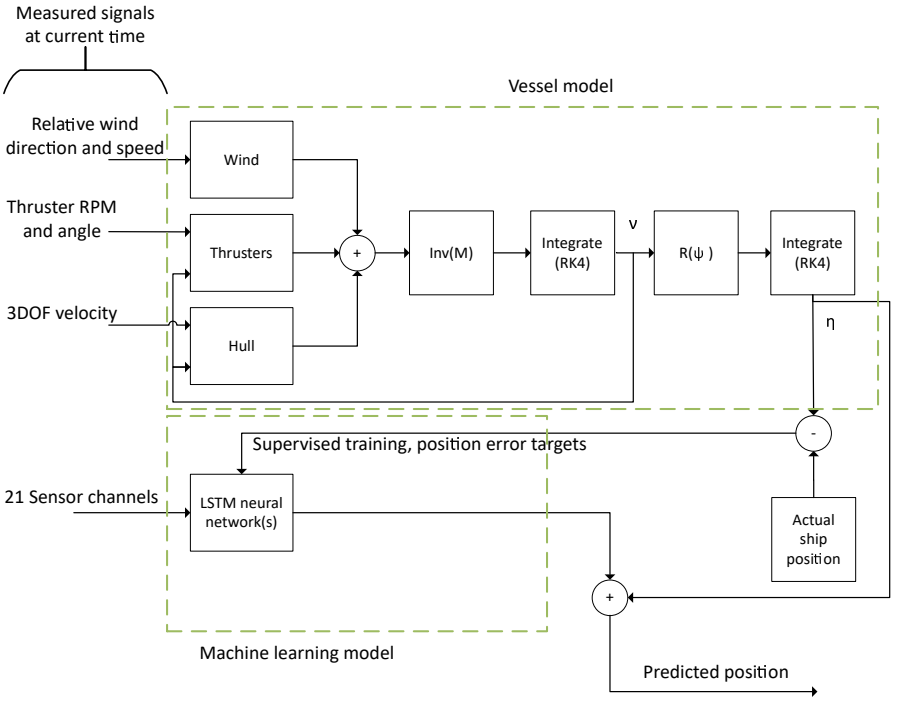


Figure 4.3: The structure of the hybrid model that predicts trajectories during docking of the R/V Gunnerus.

blue line). A significant reduction in position prediction error is observed for the hybrid predictor. Mainly this is attributed to the lack of the measured ship speed through water (STW) and the unknown future commands. The future commands causes alterations in the future trajectory that are not represented in the input of the kinetic model predictor.

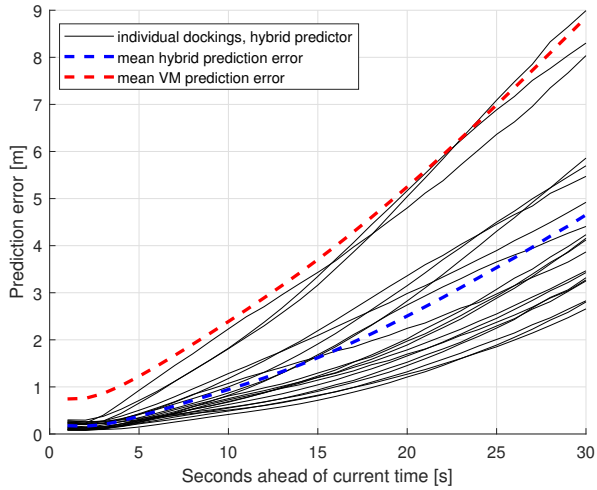


Figure 4.4: The average position prediction error of the vessel model (VM, kinetic model) predictor by itself (dashed red line) and the hybrid predictor (dashed blue line) over the 20 test sets and the prediction horizon (1-30 seconds). The solid black lines represent the average position prediction error of the 20 individual docking operations included in the test set for the hybrid predictor.

4.2.2 Hybrid DP failure predictor

As was the case for the work presented in Section 4.2.1, the parallel hybrid predictor scheme was applied in paper IV. The procedures and results of which will be presented in this section.

Where the hybrid predictor in Section 4.2.1 predicted multiple future positions at each execution, the hybrid predictor of this section predicts one step ahead (see Fig. 4.5). This is similar to how the kinetic model predictor would function if e.g. a forward Euler integration scheme was applied with a step length equal to that of the sampling rate of 1 Hz. Also the LSTM layer, which carries a memory of the previously seen data, was changed in favor of an MLP structure.

The aim of the NN predictor is to output corrections in terms of the acceleration discrepancy, $\Delta\dot{\boldsymbol{\nu}}_{t+1}$. At each index in the prediction interval the resulting position in the NED frame is obtained through Runge-Kutta (RK) integration and a rotation due to the predicted heading angle. Applying an MLP network effectively creates a mapping between discrete input vectors and the predicted acceleration error predicted by the kinetic model predictor.

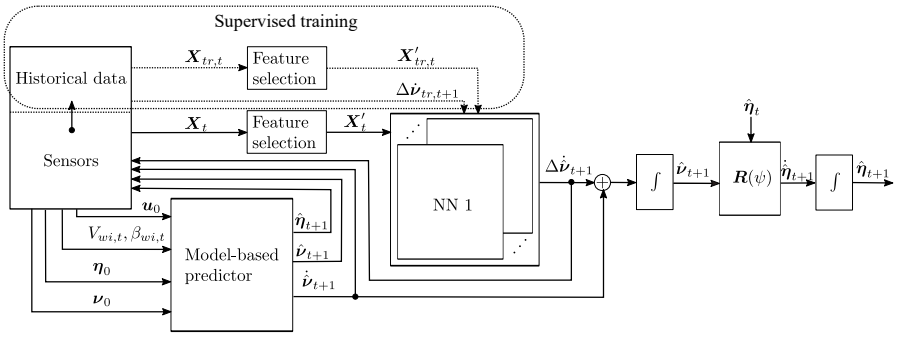


Figure 4.5: The components of the hybrid predictor that predicts trajectories in the power failure case.

Input data

For this particular case, feature selection was not performed using automated ranking or selection methods. Instead a manual selection was performed based on the specific problem. As we opted for an iterative approach, features are required to be dynamic throughout the prediction interval such that the predicted value may attain varying numerical values. If the inputs at the time of prediction was applied for all future one-step predictions the output would flat-line.

This notion drastically limits the potential features to those that are propagated by the kinetic model and, potentially, the prediction of the network. The latter results in an AR model, which is demonstrated by the connection between the output of the *NN* block to the *Sensors* block of Fig. 4.5. AR models are frequently applied in time series modelling problems, often when long-term dependencies are not dominant. An AR structure was selected and seemed a good fit for the rapidly changing dynamics of the acceleration signal. Additionally, the ship kinetic model velocity and acceleration predictions were included as inputs. All signals contain three dimensions; surge, sway and yaw, which lead to 9 input features and 3 output features.

Model structure

The design of the NN acceleration error predictor is similar to the one presented in the *Model selection* paragraph of Section 4.2.1. It applies an ensemble of networks that predicts the 3 DOF acceleration. Due to the reduced prediction interval and faster dynamics of acceleration signals compared to positions, the inherent memory of recurrent networks were considered less important. Thus, feedforward layers replaced the LSTM layers in each NN. Hyperparameters such as learning rate, number of layers, number of nodes in each layer, activation functions and regularization methods were manually searched to obtain a model that performed satisfactorily.

Results

The R/V Gunnerus, described in Section 2.1.3 served as the experimental platform for the validation of the prediction approach of paper IV. A power failure test was conducted

as described in Section 4.1.2. When initiating the DP mode, some 50 minutes prior to executing the simulated power failure, the ship operator judged the sea state to be around 2 on the Beaufort scale.

Upon arriving at the location, outside the port of Trondheim, Norway, the vessel was put in the DP mode. After 50 minutes of stationkeeping and low-speed maneuvering the simulated power failure was initiated. These 50 minutes of data (3000 samples), made up the dataset used for training. Having training data that is sampled just prior to the failure and at the same location is beneficial. It ensures that the vessel dynamics have not changed due to e.g. loading conditions and that the current and wave conditions are similar.

Fig. 4.6 shows the performance of the proposed hybrid predictor against the pure kinetic model predictor detailed in Section 2.3.1. The predicted accelerations are integrated once to obtain the velocity. The red line in each of the three subplots of the figure shows the evolution of the measured velocity throughout the first 60 seconds of drifting freely. The blue and yellow lines are the velocities predicted by the hybrid- and kinetic model predictors, respectively, starting at three seconds after the failure. Due to the sensitivity of the kinetic model to perturbations in the initial heading and velocities, a 3-second sliding window low pass filter was applied to these states prior to using them in the kinetic model predictor.

The top plot of Fig. 4.6 indicates a large surge velocity prediction error for the kinetic model predictor. Ocean current velocity is not measured on the R/V Gunnerus. Since the velocity of the vessel relative to the water is an input parameter of the kinetic model, the prediction error may come from either model discrepancies or from the unmeasured forces induced by waves and ocean current. The hybrid surge velocity prediction displays a significantly reduced prediction error. Sway- and yaw velocity predictions are also improved by the hybrid predictor, although the improvement versus the kinetic model predictor is less.

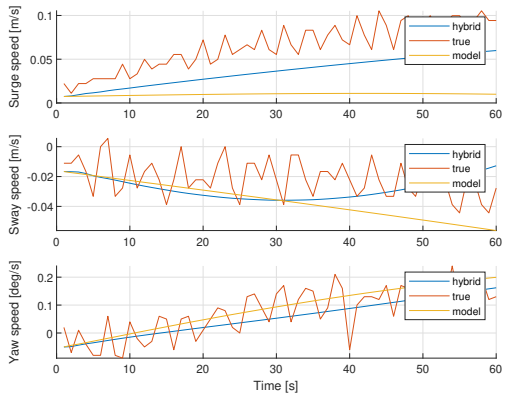


Figure 4.6: The velocities related to the first prediction interval starting from three seconds after the point of failure.

The first five trajectories for each predictor are plotted in Fig. 4.7. It illustrates their performance and repeatability. It should be mentioned that a large portion of the

trajectory error of the kinetic model predictor stems from the poor surge axis predictions. This leads to a trajectory that is more influenced by the sway velocity predictions. Nevertheless, the positive impact of the NNs in the hybrid predictor is clear.

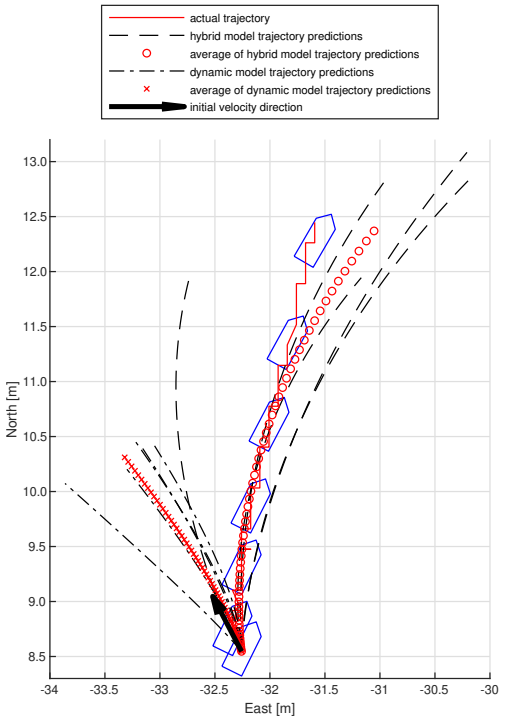


Figure 4.7: Trajectories of the ship while drifting. Each black line (dash or dash-dot) corresponds to a 60 second prediction that originates at t=[3,7] for a failure at t=0. The blue ship frames, plotted every 10 seconds, indicate the actual heading angle of the ship.

4.3 Chapter summary

Applying a purely kinetics-based predictor calls for accurate model parameters and measurements of all the required input states. This requires an extensive system identification procedure and adds cost due to instrumentation. In the cases presented in this chapter the instrumentation deficiency comes mainly from not having measurements of the vessel's speed through water. It was shown that implementing a parallel hybrid predictor significantly improves trajectory predictions in this case. Pinpointing the cause of the kinetic model predictor's error (model parameters or lack of measurements), and thereby what prediction error the NNs of the hybrid predictor mitigate, would require further instrumentation of R/V Gunnerus.

Although comparisons were not made relative to a purely data-based predictor, it is believed that these will perform well if given a sufficient amount of data. The key word being *sufficient*. The data-based predictor would have to predict the complete

state (either acceleration, velocity or position), involving more complex dynamics. This would require a more expressive model (more parameters), which leads to the need for more input data.

5 Case study: NN control allocation

Two research items are included in this thesis that describe control allocation using neural networks: papers I and V. They contribute to fulfill RO3 and the related RQ6. Control allocation is an integral part of ship operations that require low-speed maneuvering and stationkeeping. These operations involve ships that are over-actuated (redundancy concerns and maneuvering ability), and thus the task of distributing the force contribution of each thruster is not trivial; more than one set of commands result in the same generalized force. The proposed NN allocation scheme offers fixed execution speed at the cost of an offline initial NN training period.

A preliminary study on the use of NNs in control allocation was conducted in paper I. Here, the simulation platform of Section 2.1.1 was used both to generate training data and to test the allocator. A fixed procedure was applied to manually issue thruster commands taking into account the known command limits and rate limits. By sampling the force output and the corresponding thruster command, a feedforward NN could be trained that mapped generalized force to thruster commands. The generalized force vector, $\boldsymbol{\tau}$, is the control force vector along the surge and sway axes and moment about the yaw axis of the ship. This approach faces challenges related to the procurement of training data and the requirement of having thruster force measurements.

To overcome these challenges, and also directly allow for primary/secondary constraint considerations, a new design was proposed in paper V. Here, the data generation step and network structure and training was modified. The rest of this chapter will therefore be devoted to this approach.

5.1 Data

Control allocation methods based on online optimization routines, such as SQP, perform operations on a single sample. Potentially they also require knowledge about the previous allocated values. NNs on the other hand need to perform model adaptation, based on some data related to the modelled process, in order to be useful. Here, this data is synthetically generated. Based on the knowledge of the type of thrusters (rotatable or not), their location and inherent rate- and maximum force constraints, training data within a certain command range is generated. A description of the data generation process is given in Section 2.4.2.

5.2 Loss functions

Loss functions describe how well an NN models a given problem. Based on the numerical value of the loss function output, parameters of the NN are adapted. The objective is to minimize the numerical value of the loss function. So, by constructing loss functions

that dictate what a good prediction is, the behaviour of the NN may be shaped. This is how the constraints, typically evaluated in objective functions for quadratic programming approaches, have been transferred to NNs in paper V. Section 2.4.3 outlined the general purpose of the applied loss functions. For a more detailed explanation of the loss functions, please refer to paper V in the Appendix.

5.3 Results

Fig. 2.7 showed the overall structure of the autoencoder network used in paper V, and Section 2.4.1 gave a description of the general structure. A manual search resulted in the given structure. Layers 1-2 and 4-5 contain 64 nodes each and are LSTM layers. Layer 3 and 6 contain 5 and 3 outputs, respectively, and are regular feedforward layers.

The experiment was conducted in the simulator described in Section 2.1.2 using the R/V Gunnerus. Its thruster layout is given in Fig. 2.2. Two individual tests were performed to gauge the NN allocator performance in relation to a fixed-angle allocator described in [4]: low-speed maneuvering through a rectangular track with a constant wind disturbance and stationkeeping with a varying wind disturbance.

Rectangular path tracking

A path tracking scenario was constructed that requires maneuvers where the ship moves along/about more than one of its axes. The procedure detailed in paper V is re-iterated here for readability.

A: Initiate the vessel at a heading of 0 degrees at location (0 m north, 0 m east).

B: Move 20 m straight north. Start time: 0 s.

C: Move 20 meters straight west. Start time: 250 s.

D: While at location (20 m north, 20 m east), rotate the vessel to achieve a heading of 315 degrees. Start time: 450 s.

E: While maintaining a heading of 315 degrees, move 20 m south. Start time: 650 s.

F: Move 20 m east while also rotating to a heading of 0 degrees. Start time: 850 s.

The simulation components needed to enable the simulated test are shown in Fig. 5.1. The *Reference generator* outputs smooth position and heading references as well as smooth velocity references to the *Motion Controller*. It is based on the controller described in [3], which outputs the desired generalized force, $\boldsymbol{\tau}$, to the *Control Allocator*. The control allocator's output, $\boldsymbol{u}$, consists of the desired force for each thruster as well as the angles of the azimuth thrusters, shown in Fig. 2.2. Angles are passed directly to the hydrodynamic model of each thruster, but the forces go to three PI controllers that output torque commands to separate electrical motors per thruster. From these motors the hydrodynamic model of each thruster receives a rotational speed command [55].

A fixed-angle control allocation method was implemented to act as a performance reference [4]. Three sets of angles for α_2 and α_3 was tested on the rectangular path tracking test. Based on the power consumption and tracking error, the fixed angles $\alpha_2 = -45$ and $\alpha_3 = 45$ were selected.

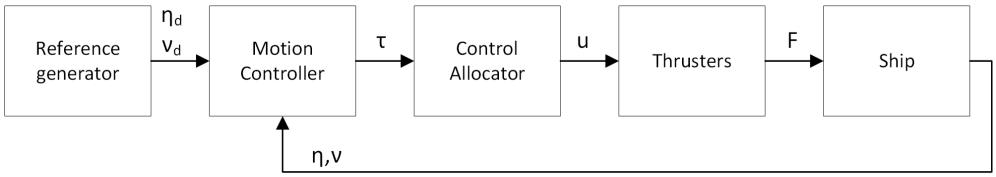


Figure 5.1: Simulation components related to the NN control allocation case study.

Fig. 5.2 shows the path of the vessel for each of the two allocation methods. They are virtually identical. This means that they are both able to issue commands that ensures a close match between the motion controller request and the generalized force produced jointly by the thrusters.

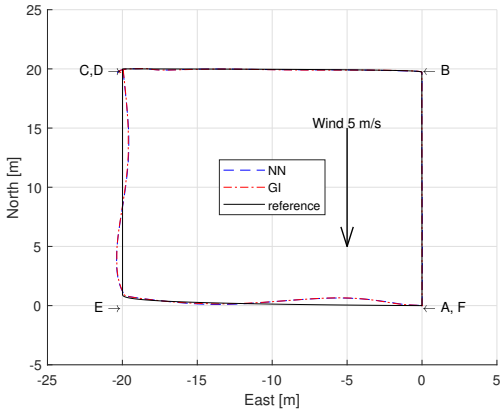


Figure 5.2: Path taken for both the neural allocator and the GI allocator. The latter applies fixed azimuth angles of $\alpha_2 = -45$ and $\alpha_3 = 45$ degrees.

The commands issued by both allocators are given in Fig. 5.3. Since the azimuth angles of the NN allocator were allowed to vary, which allows the azimuth thrusters to operate in more efficient directions, the maximum forces are slightly lower. In addition, the prescribed test offers ample time for the vessel to transition between the waypoints given in Fig. 5.2. This leads to low vessel speeds and thereby low thruster force commands. As a result of this the force magnitudes and force/angle rates of the thrusters kept well within the constraints set for this test.

Some of the loss functions (L_2, L_3 and L_5) are active only if the constraint they penalize is breached. They are designed in this way to not impose restrictions within e.g. the allowable force/angle rate range. Although they are not active during this test, they contribute to shape the weights of the NN allocator during training since the training dataset specifically introduces unattainable generalized force requests. Thus, Fig. 5.4 only displays the losses that are permanently active. It is clear that the power loss, L_4, dominates the overall loss, L. A relatively low L_1 value indicates that the autoencoder estimates the generalized force request well which suggests that the training procedure

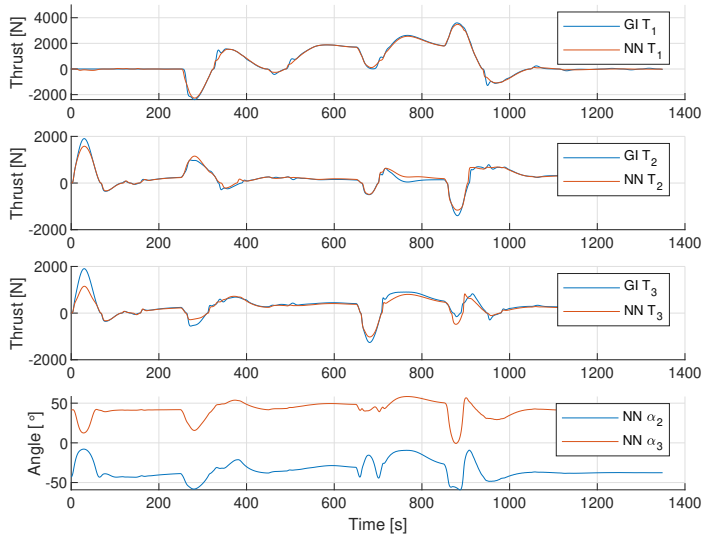


Figure 5.3: Force and azimuth angle commands issued by both allocators. Note that the bottom plot only contains the azimuth angles of the neural network allocator.

has been successful.

Power consumption is an important metric for any vessel as it directly impacts operational and environmental costs. A comparison was made in Fig. 5.5 between the fixed-angle and NN allocator's power consumption. The NN allocator has the option of rotating the azimuth thrusters towards more efficient directions. As expected this leads to a reduced power consumption.

The execution speed of the presented approach was measured to 6 ms. It is therefore slower than the SQP method of [30] (0.385 ms, reported by [56]), and faster than the MPC method of [56] (10 ms). The non-iterative execution of the presented allocator means that the execution time is not dependent on the inputs.

Stationkeeping

To highlight the impact of varying power loss scaling, a stationkeeping test was run. Three allocators were trained, where the value for k_4 was assigned the values 1×10^{-7}, 3×10^{-7} and 5×10^{-7}. If all other parameters of the network and its training procedure are left unchanged, the expectation is that a larger k_4 leads to a reduction in power consumption. The simulation test was designed to excite the force request output by the motion controller. By changing the direction of a fixed-speed uniform wind field this was accomplished. Initially the wind was coming from the east. After a 5 second transition period the wind was coming from northeast. Fig. 5.6a and 5.6b display a section of the test centered around the wind change event given at t=150 s.

Fig. 5.6a show the overall loss and the power consumption of the thrusters. From this figure the sensitivity of the overall loss to the power loss, L_4, is clear. In addition, the effect of penalizing the use of thrust is a reduction in the overall power consumption,

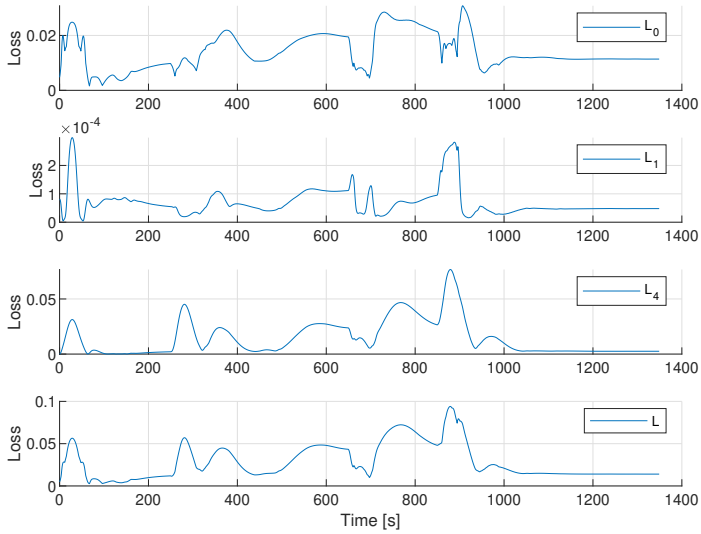


Figure 5.4: Losses incurred by the neural allocator during the 4-corner test.

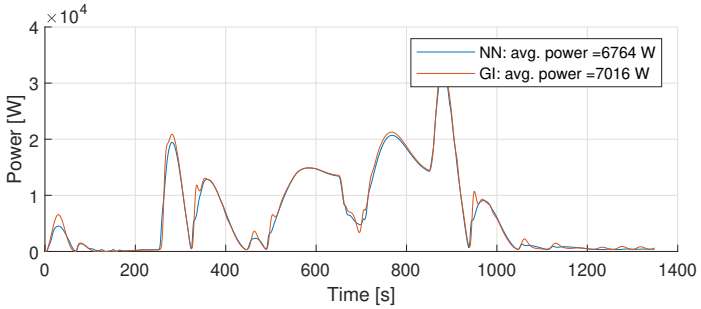
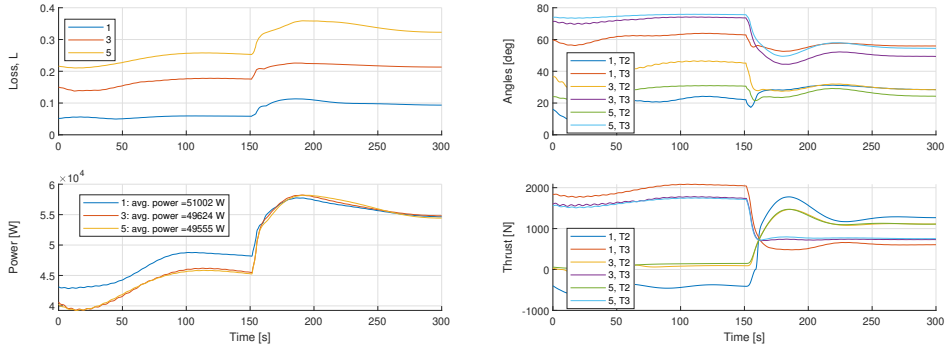


Figure 5.5: Power consumption of the GI allocator and the neural allocator during the 4-corner test.

as one would expect. An indirect effect of increasing the power loss scaling factor is seen in Fig. 5.6b, which displays the commands issued to the azimuth thrusters. As L_4 increases, the azimuth angles attain more effective directions. This is a result of the primary objective of the allocator; to make the thrusters jointly produce the requested force. A downside to this coupled effect is added difficulty in selecting appropriate loss scaling factors, k_0-k_5.



(a) Power consumption (bottom) and overall loss (top)

(b) Thrust (bottom) and azimuth angles (top)

Figure 5.6: Performance of the NN control allocator during the stationkeeping test. The legend entries indicate: 1: $k_4 = 1 \times 10^{-7}$, 3: $k_4 = 3 \times 10^{-7}$, 5: $k_4 = 5 \times 10^{-7}$. T_2 and T_3 indicate the azimuth thrusters.

5.4 Chapter summary

A NN allocator enforcing constraints similar to that of existing solutions for control allocation of over-actuated ships have been tested in this chapter. Given the offline training scheme applied, the method is applicable for a fixed set of thrusters. That is, if a thruster is deactivated during maneuvering, either by intention or by a fault in the thruster, the model is invalidated. Furthermore, the case study applied a bounded range of motion controller force requests during training. This provided sufficient control forces for the presented case study, but higher environmental loads and maneuvering dynamics would require an extension of the motion controller force requests experienced during training. Effects of this are an exponential increase in training data samples, which requires extra time spent training the model, as well as a reconfiguration of the complexity of the model.

6 Conclusion and further work

This thesis revolves around motion prediction and control allocation for ships. The majority of the research work was allocated to explore ship motion prediction approaches and identify operations that could benefit from this. In doing so, pure ML models, a parallel hybrid model and purely kinetics-based predictors were explored seeking to obtain RO1 and RO2. Current ship motion prediction methods mainly adopt either the data-based approach or the kinetic/model-based approach. The research put forward in this thesis seeks to bridge this gap and leverage the complementary features of the two prediction approaches. Case studies described in Chapter 4 confirm the increased prediction accuracy observed for the hybrid prediction models relative to the kinetics-based models. Hybrid modelling approaches from other domains were used as inspiration for the presented hybrid parallel approach.

Using ML to represent the mapping between motion controller requests and individual thruster commands was also explored to obtain RO3. The intuition here was that a trained network would be beneficial in terms of offering a non-iterative allocator. A key trait of such an allocator, relative to current state-of-the art methods, is a fixed-time execution. Functionality similar to the existing approaches was ensured through loss functions imposed during the training of the ML model.

6.1 Summary of contributions

- Present operations where short-term, accurate trajectory predictions may aid ship operators. Three operations were identified: Dead reckoning during a GNSS signal loss while performing stationkeeping, trajectory prediction during docking maneuvers and trajectory prediction while experiencing a power loss during stationkeeping.

- Propose a data-driven modelling scheme for DR.

- Propose two variants of hybrid predictors that can merge knowledge from a predetermined, physics-based ship motion model and data sampled from the ship. Prediction horizons of 30 and 60 seconds were considered for the docking and power failure scenarios, respectively.

- Outline a non-iterative approach for ship control allocation using NNs. Simulation scenarios, where the proposed allocator was packaged in an FMU, were constructed to test the performance.

6.2 Summary of publications

Paper I presents an initial approach to control allocation using a feedforward NN. Based on thruster command data generated in a simulator, two allocator instances were created. The possibility of rotating thrusters and power minimization was not considered. However, magnitude and rate constraints were, and the allocator displaying adherence to these constraints applied a nonlinear AR structure.

Paper II presents a data-driven approach to DR for a ship. A recurrent network model was compared against a feedforward network and a KF. The latter embedded a linearized maneuvering model of the simulated ship. Evaluating the position estimation capability on a one minute period of GNSS signal loss showed that the recurrent model provided the best performance.

Paper III contains the initial work related to combining models from the physics-based and the data-based modelling domains. The parallel approach is investigated in which the trajectory predictions of the kinetic model of the RV Gunnerus is corrected by a recurrent network predictor. Here, data from the RV Gunnerus, sampled during a one-year period, was applied to train the hybrid predictor and test the overall predictive performance across several docking locations on the west coast of Norway.

Paper IV builds on the approach presented in paper III. However, applying a power failure scenario motivated the use of a one-step iterative prediction scheme. While the predictor of paper III outputs the trajectory of the vessel on the entire prediction interval at once, this predictor outputs a one-second prediction and is re-iterated to provide the complete trajectory prediction. In doing so, the NN part of the predictor acts as a mapping between the predictions of the kinetic model and the actual trajectory.

Paper V is a continuation of the work presented in paper I. It relaxes the requirement of having data sampled from a (simulated) ship. Based on the specific thruster configuration, and a defined per-thruster operational range, synthetic motion controller requests are created to perform training in an Autoencoder-like network. User-defined constraints are also included to provide functionality similar to that of established ship control allocation methods. The allocator was tested in a simulator to gauge its ability for low-speed trajectory tracking and stationkeeping.

6.3 Future work

This thesis has focused on ship motion prediction and control allocation. A recurring theme has been the use of NNs to partly or fully model the functional relationships presented in each of the five research items. The below bullet points provide suggestions for how the presented research may be extended.

- Designing an interface that conveys predictions to a ship operator, shown as "Decision support" in Fig. 1.1, was not performed during this Ph.D. project. This would be an important step to gauge the effectiveness and usability of such an approach. It could also foster continual (online) training of the deployed models.
- Automatic decision-making, enabling autonomous navigation of ships, will become increasingly important in the years and decades to come. Autonomous ship navigation transfers a significant part of the responsibility of ensuring safe navigation from the on-site ship operator to the autonomous navigation system. Motion prediction

methods, which support automatic decision-making, should therefore convey their confidence in the predicted trajectory being correct. This is a challenge considering the operational environment of ships and the multitude of factors that dictate their future motion. Developing uncertainty-aware motion prediction methods is therefore seen as a key step towards autonomous ship navigation.

- In general, the performance of data-driven models are dependent on the selection of inputs, the quality and amount of data available for model training and selection of hyperparameters. Performing these operations on a wider search space requires significant computational resources, but would likely return higher performing model configurations (model parameters and input feature subsets).
- The NN control allocator presented in this thesis enforces constraints according to Section 2.4.3. However, they are not enforced in a hard way. This means that the thruster force and rotational constraints may be violated. Causes leading to such violations may be improper training or motion controller requests that are not within the numerical range experienced during training. Partly, these considerations have been explored and addressed in a Master thesis project within the Department of Ocean Operations and Civil Engineering at NTNU Aalesund. It would also be beneficial to investigate the behaviour of the allocator during failures in thrusters. For such cases an allocator discarding the failed thruster may be trained and a transition between the original allocator and the reduced-output allocator may be performed.

References

[1] M. R. Endsley, "Proceedings of the Human Factors and Ergonomics," in *Proceedings of the Human Factors Society*, pp. 97–101, 1988.

[2] M. Laurinen, "Remote and Autonomous Ships: The next steps," tech. rep., Advanced Autonomous Waterborne Applications (AAWA) Initiative, 2016.

[3] A. J. Sørensen, "A survey of dynamic positioning control systems," *Annual Reviews in Control*, vol. 35, no. 1, pp. 123–136, 2011.

[4] T. I. Fossen, *Handbook of Marine Craft Hydrodynamics and Motion Control*. John Wiley and Sons Ltd., 2nd ed., 2021.

[5] S. Küchler, T. Mahl, J. Neupert, K. Schneider, and O. Sawodny, "Active control for an offshore crane using prediction of the vessels motion," *IEEE/ASME Transactions on Mechatronics*, vol. 16, no. 2, pp. 297–309, 2011.

[6] X. Yang, H. Pota, M. Garratt, and V. Ugrinovskii, "Ship Motion Prediction for Maritime Flight Operations," in *Proceedings of the 17th World Congress The International Federation of Automatic Control*, vol. 41, pp. 12407–12412, IFAC, 2008.

[7] D. C. Psichogios and L. H. Ungar, "A hybrid neural network-first principles approach to process modeling," *AIChE Journal*, vol. 38, no. 10, pp. 1499–1511, 1992.

[8] P. W. J. 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.

[9] A. Karpatne, W. Watkins, J. Read, and V. Kumar, "Physics-guided Neural Networks (PGNN): An Application in Lake Temperature Modeling." http://arxiv.org/abs/1710.11431, 2017.

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

[11] M. L. Thompson and M. A. Kramer, "Modeling chemical processes using prior knowledge and neural networks," *AIChE Journal*, vol. 40, no. 8, pp. 1328–1340, 1994.

[12] U. Forssell and P. Lindskog, "Combining Semi-Physical and Neural Network Modeling: An Example ofIts Usefulness," in *IFAC Proceedings Volumes*, vol. 30, pp. 767–770, Elsevier, 1997.

[13] S. O. Halstensen, L. Vasilyev, V. Zinchenko, and Y. Liu, "'Next minutes' ocean waves and vessel motion predictions for more efficient offshore lifting operations," *SNAME Maritime Convention 2020, SMC 2020*, pp. 1–11, 2020.

[14] L. P. Perera and C. Guedes Soares, "Ocean Vessel Trajectory Estimation and Prediction Based on Extended Kalman Filter," in *ADAPTIVE 2010, The Second International Conference on Adaptive and Self-Adaptive Systems and Applications*, pp. 14–20, 2010.

[15] L. P. Perera, "Navigation vector based ship maneuvering prediction," *Ocean Engineering*, vol. 138, pp. 151–160, 2017.

[16] M. Triantafyllou, M. Bodson, and M. Athans, "Real time estimation of ship motions using Kalman filtering techniques," *IEEE Journal of Oceanic Engineering*, vol. OE-8, no. 1, pp. 9–20, 1983.

[17] I. Yumori, "Real Time Prediction of Ship Response to Ocean Waves Using Time Series Analysis," in *Oceans 81*, pp. 1082–1089, 1981.

[18] G. D. Masi, F. Gaggiotti, R. Bruschi, and M. Venturi, "Ship motion prediction by radial basis neural networks," in *2011 IEEE Workshop On Hybrid Intelligent Models And Applications*, pp. 28–32, 2011.

[19] J. Yin, N. Wang, and A. N. Perakis, "A Real-Time Sequential Ship Roll Prediction Scheme Based on Adaptive Sliding Data Window," *IEEE Transactions on Systems, Man, and, Cybernetics: Systems*, pp. 1–11, 2017.

[20] T. Praczyk, "Using evolutionary neural networks to predict spatial orientation of a ship," *Neurocomputing*, vol. 166, pp. 229–243, 2015.

[21] 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.

[22] M. W. Li, J. Geng, D. F. Han, and T. J. Zheng, "Ship motion prediction using dynamic seasonal RvSVR with phase space reconstruction and the chaos adaptive efficient FOA," *Neurocomputing*, vol. 174, pp. 661–680, 2016.

[23] M. W. Li, J. Geng, W. C. Hong, and L. D. Zhang, "Periodogram estimation based on LSSVR-CCPSO compensation for forecasting ship motion," *Nonlinear Dynamics*, vol. 97, no. 4, pp. 2579–2594, 2019.

[24] W. C. Hong, M. W. Li, J. Geng, and Y. Zhang, "Novel chaotic bat algorithm for forecasting complex motion of floating platforms," *Applied Mathematical Modelling*, vol. 72, pp. 425–443, 2019.

[25] 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 Twenty-fifth (2015) International Ocean and Polar Engineering Conference*, pp. 59–66, 2015.

[26] T. A. Johansen and T. I. Fossen, "Control allocation - A survey," *Automatica*, vol. 49, no. 5, pp. 1087–1103, 2013.

[27] F. Scibilia and R. Skjetne, "Constrained control allocation for vessels with azimuth thrusters," in *IFAC Conference on Manoeuvring and Control of Marine Craft, 2012*, pp. 7–12, IFAC, 2012.

[28] M. Rindaroey and T. A. Johansen, "Fuel optimal thrust allocation in dynamic positioning," in *IFAC Conference on Control Applications in Marine Systems*, pp. 43–48, IFAC, 2013.

[29] A. Veksler, T. A. Johansen, R. Skjetne, and E. Mathiesen, "Thrust Allocation With Dynamic Power Consumption Modulation for Diesel-Electric Ships," *IEEE Transactions on Control Systems Technology*, vol. 24, no. 2, pp. 578–593, 2016.

[30] T. A. Johansen, T. I. Fossen, and S. P. Berge, "Constrained Nonlinear Control Allocation With Singularity Avoidance Using Sequential Quadratic Programming," *IEEE Transactions on Control Systems Technology*, vol. 12, no. 1, pp. 211–216, 2004.

[31] D. W. Zhao, F. G. Ding, J. F. Tan, Y. Q. Liu, and X. Q. Bian, "Optimal thrust allocation based GA for dynamic positioning ship," in *2010 IEEE International Conference on Mechatronics and Automation, ICMA 2010*, pp. 1254–1258, IEEE, 2010.

[32] D. Wu, F. Ren, and W. Zhang, "An energy optimal thrust allocation method for the marine dynamic positioning system based on adaptive hybrid artificial bee colony algorithm," *Ocean Engineering*, vol. 118, pp. 216–226, 2016.

[33] O. J. Sørdalen, "Optimal thrust allocation for marine vessels," *Control Engineering Practice*, vol. 5, no. 9, pp. 1223–1231, 1997.

[34] V. Hassani, A. Ross, Ø. Selvik, D. Fathi, F. Sprenger, and T. E. Berg, "Time Domain Simulation Model For Research Vessel Gunnerus," in *International Conference on Ocean, Offshore and Arctic Engineering*, pp. 1–6, 2015.

[35] S. Sadjina, L. T. Kyllingstad, M. Rindarøy, S. Skjong, V. Esøy, and E. Pedersen, "Distributed co-simulation of maritime systems and operations," *Journal of Offshore Mechanics and Arctic Engineering*, vol. 141, no. 1, 2019.

[36] O. N. Smogeli, *Control of Marine Propellers: From Normal to Extreme Conditions*. PhD thesis, Norwegian University of Science and Technology, 2006.

[37] I. Guyon and A. Elisseeff, "An Introduction to Variable and Feature Selection," *Journal ofMachine Learning Research*, pp. 1157–1182, 2003.

[38] F. Hutter and K. Leyton-brown, "An Efficient Approach for Assessing Hyperparameter Importance," in *Proceedings of the 31 st International Conference on Machine Learning*, 2014.

[39] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning Representations by Back-Propagating Errors," *Nature*, vol. 323, pp. 533–536, 2019.

[40] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[41] R. Jozefowicz, W. Zaremba, and I. Sutskever, "An empirical exploration of Recurrent Network architectures," *32nd International Conference on Machine Learning, ICML 2015*, vol. 3, pp. 2332–2340, 2015.

[42] Y. Bengio, P. Simard, and P. Frasconi, "Learning Long-Term Dependencies with Gradient Descent is Difficult," *IEEE Transactions on Neural Networks*, vol. 5, no. 2, pp. 157–166, 1994.

[43] K. Greff, R. K. Srivastava, J. Koutník, B. R. Steunebrink, and J. Schmidhuber, "LSTM: A Search Space Odyssey," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 28, no. 10, pp. 2222–2232, 2017.

[44] M. Raissi, Z. Wang, M. S. Triantafyllou, and G. E. Karniadakis, "Deep learning of vortex-induced vibrations," *Journal of Fluid Mechanics*, vol. 861, pp. 119–137, 2019.

[45] J. Willard, X. Jia, S. Xu, M. Steinbach, and V. Kumar, "Integrating Physics-Based Modeling with Machine Learning: A Survey." http://arxiv.org/abs/2003.04919, 2020.

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

[47] T. Suzuki, M. Sugiyama, T. Kanamori, and J. Sese, "Mutual information estimation reveals global associations between stimuli and biological processes," *BMC Bioinformatics*, vol. 10, no. SUPPL. 1, pp. 1–19, 2009.

[48] J. Ding, V. Tarokh, and Y. Yang, "Model Selection Techniques: An Overview," *IEEE Signal Processing Magazine*, vol. 35, no. 6, pp. 16–34, 2018.

[49] S. Bouktif, A. Fiaz, A. Ouni, and M. A. Serhani, "Optimal deep learning LSTM model for electric load forecasting using feature selection and genetic algorithm: Comparison with machine learning approaches," *Energies*, vol. 11, no. 7, 2018.

[50] J. Bergstra, R. Bardenet, Y. Bengio, and B. Kégl, "Algorithms for hyper-parameter optimization," in *Advances in Neural Information Processing Systems 24: 25th Annual Conference on Neural Information Processing Systems 2011, NIPS 2011*, pp. 1–9, 2011.

[51] J. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization," *Journal of Machine Learning Research*, vol. 13, pp. 281–305, 2012.

[52] J. Snoek, H. Larochelle, and R. Adams, "Practical Bayesian Optimization of Machine Learning Algorithms.," *Advances in neural information processing systems*, pp. 2951–2959, 2012.

[53] H. Huan, W. Wan, C. We, and Y. He, "Constrained Nonlinear Control Allocation based on Deep Auto-Encoder Neural Networks," *2018 European Control Conference, ECC 2018*, pp. 2081–2088, 2018.

[54] F. Arditti, F. L. Souza, T. C. Martins, and E. A. Tannuri, "Thrust allocation algorithm with efficiency function dependent on the azimuth angle of the actuators," *Ocean Engineering*, vol. 105, pp. 206–216, 2015.

[55] S. Skjong, M. Rindarøy, L. T. Kyllingstad, V. Æsøy, and E. Pedersen, "Virtual prototyping of maritime systems and operations: applications of distributed co-simulations," *Journal of Marine Science and Technology*, vol. 23, no. 4, pp. 835–853, 2018.

[56] S. Skjong and E. Pedersen, "Nonangular MPC-Based Thrust Allocation Algorithm for Marine Vessels - A Study of Optimal Thruster Commands," *IEEE Transactions on Transportation Electrification*, vol. 3, no. 3, pp. 792–807, 2017.

Appendix

A

Paper I

A Neural Network Approach to Control Allocation of Ships for Dynamic Positioning

Robert Skulstad*[1], Guoyuan Li[1], Houxiang Zhang[1], and Thor I. Fossen[2]

[1]Norwegian University of Science and Technology, Department of Ocean Operations and Civil Engineering, Aalesund, Norway (e-mail: robert.skulstad, guoyuan.li, hozh @ntnu.no).
[2]Norwegian University of Science and Technology, Department of Engineering Cybernetics, Trondheim, Norway (e-mail: thor.fossen@ntnu.no).

Abstract

Dynamic Positioning (DP) of ships is a control mode that seeks to maintain a specific position (stationkeeping) or perform low-speed maneuvers. In this paper, a static Neural Network (NN) is proposed for control allocation of an over-actuated ship. The thruster force and commands are measured during a trial run of the simulated vessel to gather data for training of the NN. Then the network is trained and used to transform the virtual force commands from a motion controller into individual thruster commands. A standard Proportional Integral Derivative (PID) controller, using wave-filtered position and heading measurements, is implemented as motion controller for each Degree Of Freedom (DOF) of the ship. For a DP application the controllable DOFs are the translational motion in surge and sway directions, as well as the rotation about its up/down axis. Simulation tests were performed to verify the feasibility of this approach.

Keywords— PID controllers, Neural-network models, Dynamic positioning, Control allocation

1 Introduction

Ships that are involved in safety-critical operations related to drilling, cargo-transfer, subsea crane operations and pipe-laying typically have an extended actuator setup to allow for redundancy in case of system errors. During such operations the vessel is required to control its position and heading. This operational mode is known as Dynamic Positioning (DP), which performs stationkeeping or low-speed maneuvering. In terms of the degrees of freedom (DOF) of the vessel, DP normally controls the surge (longitudinal), sway (lateral) and yaw (rotation about the up/down axis) simultaneously. Conventional ships use either tunnel thrusters, azimuth thrusters or main thrusters for thrust generation

In order to perform DP operations a modular approach to motion control is often applied (Johansen and Fossen (2013)). A top-level motion controller converts the error between the desired state and the actual state of the ship, into a generalized force vector in surge/sway direction and moment about the yaw axis. Then a control allocation module distributes that force vector into individual thruster commands to fulfill the requirements of the motion controller.

Historically, optimization-based control allocation techniques have dominated. This allows for flexibility in terms of optimization goal (and sub-goals), motivating advances in minimum power schemes and minimization of actuator wear. Lindegaard and Fossen (2003) exploited the operation of a rudder for lateral thrust to derive an energy-efficient thrust allocation algorithm for low speed operations. An explicit two-step solution was suggested to calculate a feasible thrust vector, u. The approach was limited to one rudder at a time. Their algorithm was extended by Johansen et al. (2008) to allow any number of rudders. Magnitude and rate constraints were also accounted for. Perez and Donaire (2009) handled both magnitude constraints and rate constraints in the top-level motion controller by an anti-windup controller. By constraining the virtual control vector from the motion controller, they could perform unconstrained control allocation which was posed as an optimization problem. The solution to this unconstrained control allocation may be found in (Fossen (2011)).

*Corresponding author

Sørdalen (1997) used filtering techniques to tackle the problem of azimuth angle rate constraints for rotatable thrusters. Without this constraint consideration, a singular thruster configuration may occur when rotatable thrusters are used actively. The result is failing to meet the control commands of the control law.

An adaptive genetic algorithm was used to solve the thrust allocation problem in (Zhao et al. (2010)). A fitness function was constructed based on an objective function with constraints. They considered thrust allocation for a semi-submersible rig using rotatable thrusters. Constraints considered were thruster force limits, its change rate, angular rate of thrusters and respective forbidden azimuth angles. Bui and Kim (2011) presented a control allocation scheme that involved the use of external thrust providers in the form of autonomous tugboats for ship berthing application. A constrained optimization problem was used, which they solved using a redistributed pseudo-inverse algorithm.

Chen and Jiang (2012) transformed the constrained allocation task into a convex quadratic programming problem for constrained control allocation. To solve this they applied a recurrent neural network. A neural network control algorithm was applied in (Zhang et al. (2017)) to overcome actuator gain uncertainties and to compensate for unmodelled environmental disturbances. The algorithm was tested in a simulator using six thrusters, where one of them was rotatable. Realistic environmental disturbances were applied in the simulation test.

Except for the approach described by Zhang et al. (2017), the methods described above require knowledge about the command-to-force relationship of each thruster.

Within the aerospace industry, control allocation has also received significant attention. An overview of methods used within this domain is given in Oppenheimer et al. (2006), while an evaluation of methods is given in Bodson (2002).

In this article we propose to use a neural network to obtain the mapping between the virtual generalized force, commanded by the motion controller, and the individual thruster commands. It will consider thruster rate constraints as well as limiting the maximum and minimum commands of the thrusters (magnitude constraint) for non-rotatable thrusters. Manual operation of the thrusters is used to generate the training data. Sequentially, each thruster was put through its entire operational range by first ordering a maximum command. Then, when maximum was reached, a minimum command was issued. In addition to reaching maximum/minimum thruster command values, this commands the maximum change rate as well. In this paper, no explicit support for relative weighting between thrusters exist. To achieve this, the designer would have to supply the network with a training set that reflected the desired weighting. This could be a restricted operational range for the main thrusters, resulting in lower force output by those thrusters. Two assumptions were made:

- The forces and moments imparted by each thruster on the vessel are measurable.
- The command-to-force relation gathered in the test set for calm seas is representative for command-to-force relations in all other sea states.

To validate the scheme, and for collecting training data for the NN allocation mapping, a simulated vessel will be used.

2 Control scheme

As noted by Johansen and Fossen (2013) there typically exists a hierarchy in the control system for over-actuated mechanical systems. This hierarchy allows for a modular design where each module is self-contained. Figure 1 gives an overview of the complete system used in this paper. τ_c is the commanded virtual force and moment given in the vessel frame of reference. u is the vector of individual control commands. τ is the individual horizontal plane forces imparted on the virtual ship by the thrusters. τ_{env} are the forces acting on the vessel through environmental disturbances. Included in the PID Motion controller module (section 2.2) is a wave filter, which will be described in section 2.1. The NN control allocation module of figure 1 contains both an initial training algorithm and a forward calculation of the control vector u. The latter operation is executed at each step of the complete system. An update rate of 20 Hz was used. The output of the control allocation module will be input directly to each thruster.

2.1 Wave filter

Prior to sending the relevant measurements to a control system it is customary to filter the signals to avoid compensating for high-frequency wave-induced motion (Fossen and Perez (2009)). This process is

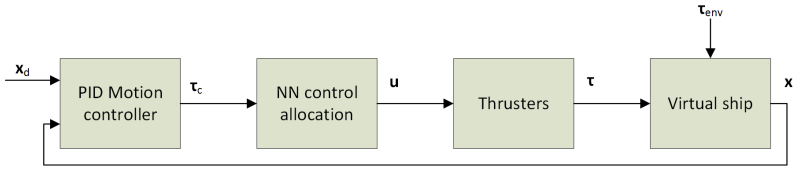


Figure 1: Overview of the proposed scheme for DP using NN allocation.

therefore often referred to as wave-filtering and is a necessary step to reduce wear and tear on actuators. Several tools can be used for this purpose including low-pass filters, Kalman filters and observers (Fossen (2011)). Low-pass filters are utilized in this stage. Although it has limited effect due to the phase lag incurred, they do not require prior knowledge of either vessel dynamics or the impact of environmental forces on the vessel.

2.2 PID motion controller

A standard PID controller was used to obtain the virtual forces necessary to converge to a given desired state. The need for integral action is evident when considering the drift force exerted on the vessel by wind, low frequency wave drift forces and ocean currents. Position measurements were given in the local North East Down (NED) frame. This made it necessary to rotate the position in the horizontal plane in order to provide error metrics aligned with the coordinate frame of the vessel. For DP operation, we define the state vector in the NED reference frame as $[N, E, \psi]^T$ and $[X, Y, \psi]^T$ for the state vector in the vessel reference frame. ψ_d is the desired heading angle of the vessel. Figure 2 shows the X and Y axis of the vessel-bound state vector.

Equation 1 yields the error signal which will act as input to the PID motion controller.

$$\begin{bmatrix} X_{err} \\ Y_{err} \\ \psi_{err} \end{bmatrix} = \boldsymbol{R}^{-1}(\psi) \left(\begin{bmatrix} N_d \\ E_d \\ \psi_d \end{bmatrix} - \begin{bmatrix} N \\ E \\ \psi \end{bmatrix} \right) \tag{1}$$

Obtaining the virtual control vector is achieved by applying a PID controller for each DOF represented in the vessel fixed state vector.

$$\tau_c = \begin{bmatrix} F_X \\ F_Y \\ M_\psi \end{bmatrix} = \begin{bmatrix} PID(X_{err}) \\ PID(Y_{err}) \\ PID(\psi_{err}) \end{bmatrix} \tag{2}$$

where $PID(e) = K_p * (e + 1/T_i * \int_0^t e + T_d * \dot{e})$ and K_p, T_i and T_d are parameters of the PID regulator subject to tuning for desired response. F_X is the surge force, F_Y is the sway force and M_ψ is the yaw moment required by the motion controller.

2.3 Neural network control allocation

A NN was used to provide the transformation between desired generalized forces (input) and the individual thruster commands (output). More specifically the learning methodology adopted was that of the Extreme Learning Machine (ELM) introduced by Guang-Bin Huang et al. (2004). Two main approaches exist with regards to adapting structure and weight parameters of the network. The most basic one involves a single, initial, calculation of weights, either using a fixed structure or adapting the structure by means of pruning/constructive methods. Alternatively, one may update both structure and weights iteratively, termed online learning, to adapt the network to time-varying properties of the process which produces the data. In this paper we adopt the first approach, fixing the structure and the weights of the network initially. Furthermore, the ELM is capable of approximating any continuous target function given proper architecture and sufficient training data. This, along with the efficiency of the algorithm, motivated the choice of using a shallow network structure as opposed to a deep neural network. The learning process of the ELM is outlined below using the same notation as Huang et al. (2011).

$$\mathbf{H} = \begin{bmatrix} \mathbf{G}(\mathbf{a}_1, b_1, \mathbf{x}_1) & \dots & \mathbf{G}(\mathbf{a}_L, b_L, \mathbf{x}_1) \\ \vdots & \dots & \vdots \\ \mathbf{G}(\mathbf{a}_1, b_1, \mathbf{x}_N) & \dots & \mathbf{G}(\mathbf{a}_L, b_L, \mathbf{x}_N) \end{bmatrix} \tag{3}$$

H is a N-by-L matrix, where N represents the number of input vectors available and L is the number of hidden layer neurons. Each node in the middle/hidden layer applies the activation function $\mathbf{G}(\mathbf{a}, b, \mathbf{x})$, where **a** is a vector of incoming weights, b is a random bias value and **x** is the input vector. A sigmoid activation function was applied for each hidden layer neuron. The only remaining unknown is the output layer weight matrix $\boldsymbol{\beta}$. This is determined using the Moore-Penrose generalized inverse as shown in equation 4. The learning scheme falls under the category of supervised learning and $\boldsymbol{T}$ therefore signifies the known true output of the system given an input vector.

$$\boldsymbol{\beta} = \boldsymbol{H}^{\dagger}\boldsymbol{T} \tag{4}$$

The condensed stepwise algorithm is given below for readability.

1. Assign random numbers to the input-to-hidden layer weights
2. Calculate the hidden layer output matrix according to equation 3
3. Determine the output layer weight parameters according to equation 4

We define two variations of the control allocation unit for comparison of performance. They differ only in terms of the structure of the NN allocator:

- Allocator 1: Only the generalized forces were applied as input to the network.
- Allocator 2: The input thrust command is augmented using the applied commands from the previous time step.

2.3.1 Allocator 1

As mentioned, the implementation of Allocator 1 used only the generalized force/moment vector issued from the motion controller as input. This is the common interface signal used by the methods referenced in the introduction. The output was the five individual command signals. To determine the proper hidden layer neuron number several different neuron numbers were tested. The neuron number corresponding to the lowest Root Mean Square Error (RMSE), based on validation by randomly selected input/target pairs of the entire training set, was selected. This approach yielded a NN structure of 3 input neurons, 20 hidden layer neurons and 6 output neurons. This mapping remained fixed for the duration of the test case given in section 3.2.

2.3.2 Allocator 2

As a means to add additional information about the rate changes experienced in the training set the input to the allocation unit was augmented. The new input consisted of the same generalized forces as in Allocator 1, in addition to the commands issued to each thruster at the previous time step. Otherwise the two allocation implementations (1 and 2) were identical. For both allocators a training set of 7800 samples was used.

3 Simulation results

Simulations were carried out in a commercial marine vessel simulator developed by the Norwegian company Offshore Simulator Centre (OSC) AS. Test case parameters are given in section 3.1.

3.1 Test case description

The vessel, shown in figure 2, was ordered to remain fixed at a position of (0,0) m in the NED reference frame and a heading angle of 0 degrees. To achieve this the vessel utilized 6 thrusters. The two stern-mounted main thrusters, which were operated in tandem, required a command of blade pitch angle (thrusters marked 5 and 6 in figure 2). Two sets of individually operated tunnel thrusters were located at the bow and stern of the vessel (thrusters marked 1-4 in figure 2). Both set of thrusters were operated using RPM commands. All thrusters were considered non-rotatable, producing positive thrust in the direction of the arrow accompanying each thruster in figure 2. Considering that the vessel is controlled in three DOFs and the number of individually operated thrusters are 5, the vessel is over-actuated. External disturbances were applied at a fixed direction, from north to south. Initially a uniform wind

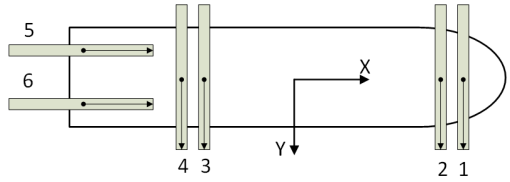


Figure 2: The vessel and its thruster layout. Note that the two leftmost thrusters (main thrusters) supply thrust in the positive/negative x-direction only.

velocity of 4 m/s affects the vessel. At the start of figure 3, a step change in wind is applied with a final value of 8 m/s. Waves generated using the JONSWAP spectrum, were applied. Both first and second order wave forces affected the vessel in the same direction as the wind and with a significant wave height of 2 meters.

In order to initialize the control allocation mapping described in section 2.3, a test set of previously recorded simulation data was used. Commanded thruster setpoint, actual thrust in Newton and the torque in Newton meter imparted by each thruster on the vessel, were recorded for the purpose of training the neural network. Each thruster was operated to its maximum and minimum command value, thus exposing its complete operational range. The model of the virtual vessel is derived from a real vessel and thus its thrusters have specific constraints with regards to command magnitude and change rate. Table 1 shows the thruster constraints.

Table 1: Specification of thruster operational constraints.

Thruster	Max	Min	Rate change
1,2	204.00 RPM	-204.00 RPM	20.40 RPM/s
3,4	276.00 RPM	-276.00 RPM	27.60 RPM/s
5,6	28.70 deg	-21.50 deg	1.44 deg/s

Although the motion controller is not the main focus of this paper, a reasonable performance is still required to provide sensible input data to the allocation unit. For this purpose a standard PID controller was applied to each of the DOFs of the vessel. To reduce the adverse effects of high frequency wave-induced motion a low-pass filter was applied to the input signals in the motion controller shown in figure 1. This was done prior to calculating the generalized forces.

3.2 Allocator 1

Figure 3 shows the performance of the complete system, which is to remain at a fixed location and heading. The red lines of figure 3 describe the performance of the system using allocator 1. The rapid change rate of this allocator allows it to compensate for the step change quickly and therefore minimize the deviation from the desired desired position. However, the red line of the middle plot of figure 3 reveals slow convergence in the East direction. The high-frequency overloaded signal seen in all plots in figure 3 is caused by the vessel interacting with waves.

Figure 4 shows the output from the allocation unit for allocator 1. It is evident that the commands generated by the allocation unit is very sensitive to variation in the generalized force command. This might suggest that further wave-filtering and controller tuning might be required for this implementation to be useful. The excessive changes in thruster commands are also visible in figure 5, showing the change in command between time steps and also the maximum change allowed for each thruster. The sampling frequency of the system was set to 20 Hz, yielding the rate limit in figures 5 and 7 based on table 1

3.3 Allocator 2

Allocator 2 is less aggressive relative to allocator 1, although the same parameters for the PID regulators have been used. The derived thruster commands change more slowly, yielding increased settling times. Similar to allocator 1, the East direction (middle plot of figure 3) response shows the slowest convergence.

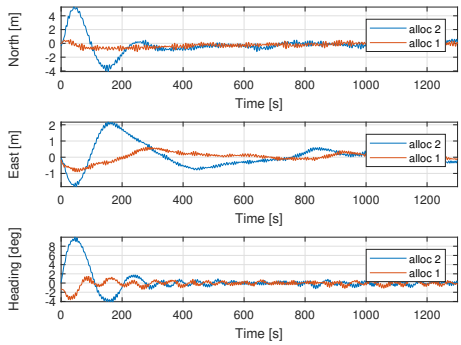


Figure 3: The position and heading of the vessel recorded during the test case run for both allocator 1 and 2.

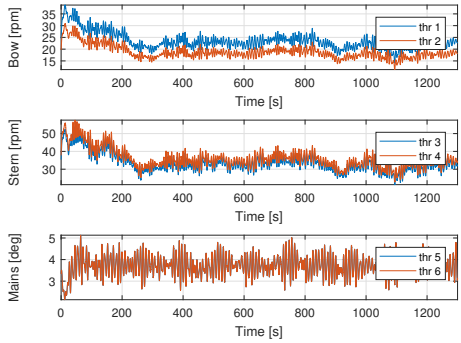


Figure 4: The commands input to each of the thrusters for the test case using allocator 1. Top: RPM commands issued to the bow tunnel thrusters. Middle: RPM commands to the stern tunnel thrusters. Bottom: Blade pitch angle commands issued to the two aft main thrusters.

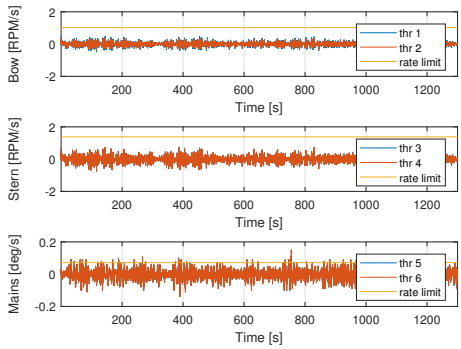


Figure 5: Shows the change rate of the commands issued to each thruster.

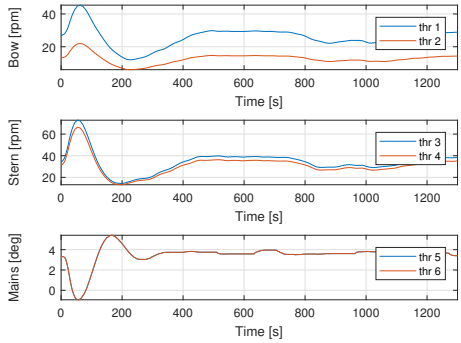


Figure 6: The commands input to each of the thrusters for the test case using allocator 2. Top: RPM commands issued to the bow tunnel thrusters. Middle: RPM commands to the stern tunnel thrusters. Bottom: Blade pitch angle commands issued to the two aft main thrusters.

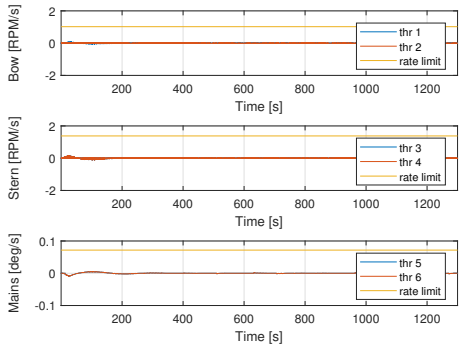


Figure 7: Shows the change rate of the commands issued to each thruster.

The commands issued by this allocator implementation result in a significant reduction in command rate of change.

Figure 7 shows the reduction in rate change due to the augmented input vector of Allocator 2. Rate constraints for all thrusters were satisfied using this implementation.

3.4 Magnitude constraint

More severe external disturbances were necessary to approach the maximum thrust limits for the thrusters. The wind and wave direction was set to act from west to east while the vessel attempted to maintain zero heading and position. Significant wave height was increased to 3 meters and the wind velocity was set to increase from 4 m/s to 20 m/s. This wind velocity increase took place at t = 2500 seconds. The result of increasing the lateral force imparted on the vessel is given in figure 8.

Figure 9 gives the corresponding thruster setpoint commands issued from the allocation unit. For this specific test Allocator 2 was used. The figure shows that the commands for thrusters 1-4 saturates, rendering the vessel under-powered in the lateral direction. Insufficient thrust force causes the vessel to drift off in the east direction (see the middle plot of figure 8) as well as failing to converge to a zero heading angle.

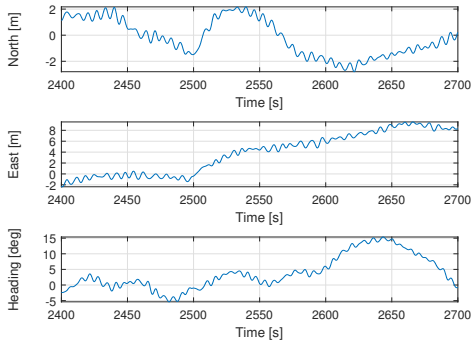


Figure 8: Recorded position and heading during the magnitude constraint test.

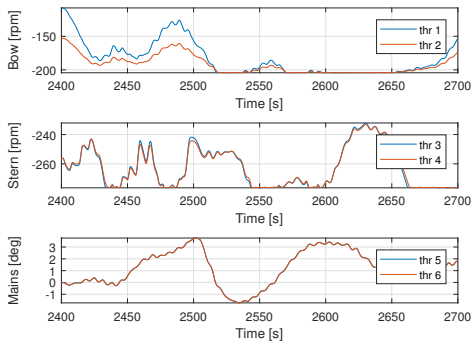


Figure 9: Displays the thruster setpoint commands for all thrusters. Top and middle plot shows saturation due to the environmental disturbances impacting the port side of the vessel.

4 Discussion

With an augmented input vector, the control allocation NN has been shown to provide constraint handling for rate change and magnitude of thrusters on a simulated vessel. Allocator 2 displays the most realistic implementation, adhering to the rate constraints present in the dataset used for training. It is therefore necessary to provide a training set that reflects the constraints of the vessel, meaning that to achieve the maximum rate change the thrusters on the vessel must be operated at its maximum rate when logging the training data. The use of thrust force measurements limits the applicability of this approach to simulators.

The test constructed to verify the ability of the NN allocator to deal with thruster magnitude constraints may not represent conditions that will be faced by a real vessel. However, it pushes the NN allocation to its thresholds, showing that it is able to constrain the commands based on the dataset used for training the NN.

Although not verified in this paper, thruster failure scenarios may be handled by a re-training step of the NN allocation mapping. For the training set and NN architecture used in this paper, the entire re-training procedure is performed in 1.3 seconds. More sophisticated functionality such as power minimization, support for thrusters rotatable in the range $[-180, 180 >$ degrees and support for forbidden zones of rotatable thrusters were not considered. They will be the aim for future improvements of the suggested approach.

5 Conclusion

In this paper we introduce a neural network thruster allocation scheme that only relies on thruster force and command measurements. Though systems onboard real vessels do not accomodate thrust force measurements, we seek here to promote an approach to control allocation that does not rely on knowledge about the specific thruster parameters. The augmented version of the neural network allocator is able to learn the relationship between the generalized thrust and thruster commands as well as handling maximum/minimum and rate constraints.

6 Acknowledgement

The authors would like to thank OSC for their support in relation to performing the simulation study.

References

Bodson, M. (2002). Evaluation of Optimization Methods for Control Allocation. *Journal of Guidance, Control, and Dynamics*, 25(4):703–711.

Bui, V. P. and Kim, Y. B. (2011). Development of constrained control allocation for ship berthing by using autonomous tugboats. *International Journal of Control, Automation and Systems*, 9(6):1203–1208.

Chen, M. and Jiang, B. (2012). Adaptive control and constrained control allocation for overactuated ocean surface vessels. *International Journal of Systems Science*, 7721(November):1–15.

Fossen, T. I. (2011). *Handbook of Marine Craft Hydrodynamics and Motion Control*. John Wiley & Sons Ltd.

Fossen, T. I. and Perez, T. (2009). Kalman Filtering for Positioning and Heading Control of Ships and Offshore Rigs: Estimating the effects of waves, wind, and current. *IEEE Control Systems*, 29(6):32–46.

Guang-Bin Huang, Qin-Yu Zhu, and Chee-Kheong Siew (2004). Extreme learning machine: a new learning scheme of feedforward neural networks. In *2004 IEEE International Joint Conference on Neural Networks*, pages 985–990.

Huang, G. B., Wang, D. H., and Lan, Y. (2011). Extreme learning machines: A survey. *International Journal of Machine Learning and Cybernetics*, 2(2):107–122.

Johansen, T. A. and Fossen, T. I. (2013). Control allocation - A survey. *Automatica*, 49(5):1087–1103.

Johansen, T. A., Fuglseth, T. P., Tøndel, P., and Fossen, T. I. (2008). Optimal constrained control allocation in marine surface vessels with rudders. *Control Engineering Practice*, 16(4):457–464.

Lindegaard, K. P. and Fossen, T. I. (2003). Fuel-efficient rudder and propeller control allocation for marine craft: Experiments with a model ship. *IEEE Transactions on Control Systems Technology*, 11(6):850–862.

Oppenheimer, M. W., Doman, D. B., and Bolender, M. A. (2006). Control allocation for over-actuated systems. In *14th Mediterranean Conference on Control and Automation, MED'06*, number 2.

Perez, T. and Donaire, A. (2009). Constrained control design for dynamic positioning of marine vehicles with control allocation. *Modeling, Identification and Control*, 30(2):57–70.

Sørdalen, O. J. (1997). Optimal thrust allocation for marine vessels. *Control Engineering Practice*, 5(9):1223–1231.

Zhang, G., Cai, Y., and Zhang, W. (2017). Robust neural control for dynamic positioning ships with the optimum-seeking guidance. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 47(7):1500–1509.

Zhao, D. W., Ding, F. G., Tan, J. F., Liu, Y. Q., and Bian, X. Q. (2010). Optimal thrust allocation based GA for dynamic positioning ship. In *2010 IEEE International Conference on Mechatronics and Automation, ICMA 2010*, pages 1254–1258.

B

Paper II

An Efficient Recurrent Neural Network for Dead Reckoning of Dynamically Positioned Ships

Robert Skulstad, Guoyuan Li, *Member, IEEE*, Thor I. Fossen, *Fellow, IEEE*, Bjørnar Vik, Houxiang Zhang, *Senior Member, IEEE*

***Abstract*—When a ship experiences a loss of position reference systems, the ship's navigation system typically enters a mode known as dead reckoning to maintain an estimate of the position of the ship. Commercial systems perform this task using a state estimator that includes mathematical model knowledge. Such a model is non-trivial to derive and needs tuning if the dynamic properties of the vessel change. To this end we propose to use machine learning to estimate the horizontal velocity of the vessel without the help of position, velocity or acceleration sensors. A simulation study was conducted to show the ability to maintain position estimates during a Global Navigation Satellite System outage. Comparable performance is seen relative to the established Kalman Filter model-based approach.**

***Index Terms*—Dead reckoning, ship motion prediction, Kalman filter, feedforward/recurrent neural network, input selection.**

I. Introduction

SYSTEM failures that occur during the performance of operations at sea that impose strict constraints on the ability of a ship to maintain position may have severe consequences. In order to mitigate the risk of failures, ships used in such operations have redundant systems. As marine operations grow increasingly autonomous and remotely operated [1], the importance of redundant systems to aid in controlling the vessel in case of failures increases. Failures that occur when a vessel is operating autonomously must be handled in a timely fashion through the issuance of a warning to a remote operator. In cases of a loss of absolute position measurements, a ship normally enters a mode known as dead reckoning (DR) to provide estimates of the vessel position without the use of external signals. Various strategies exist to provide such estimates, but the general approach is to propagate the velocity and course of the vessel from a known position [2].

With regards to the position reference used by, for example, stationkeeping motion controllers, various sensory platforms measuring the absolute or relative position may be applied, such as differential Global Navigation Satellite System (dGNSS) or radar or hydroacoustic systems [3]. While hydroacoustic-inertial navigation systems offer positioning solutions of similar quality to GNSS-inertial navigation systems, they rely on deployed seabed transponders [4]. Depending on the type of operation, this might not be a feasible strategy. The most generic and available system is therefore the dGNSS.

As positioning systems normally use signals from satellites to calculate position, there is a potential to experience both a communication dropout between the remote operator and the autonomous vessel, as well as a loss of the GNSS position reference signals. In such a case, the accuracy and long-term performance of the DR system becomes important in order to maintain an accurate estimate of the current position of the vessel.

In commercial navigation systems the Kalman Filter (KF) is often used to filter the wave-induced motion and provide estimates of the vessel velocity [5], [6]. At a minimum, position and heading measurements are input to the estimator. These measurements are combined with the linearized vessel model to provide the state estimates. Wave-filtering ensures that the oscillatory wave-induced motion does not enter the controller of the vessel causing increased fuel consumption and actuator wear [7]. The widespread use of KFs for state estimation and wave filtering makes it a natural choice as a tool for providing DR position estimates as well.

Nonlinear observers that do not require knowledge of the vessel model have also been proposed for marine vessels operating in stationkeeping mode, often referred to as Dynamic Positioning (DP). Bryne, Fossen and Johansen performed wave filtering based on the Inertial Navigation System (INS) output [8], while Rogne et al. used Inertial Measurement Unit (IMU) data for DR [9]. The difference being that the IMU provides the raw angular velocity and specific force measurements, while the INS integrates these measurements into a navigation solution in terms of a position, velocity and attitude.

In the event of a dGNSS position reference failure, the state estimator, assuming the KF is used, can make estimates based only on the vessel model [5]. The position reference failure may be caused by external factors such as loss of a direct line of sight between satellite and receiver, intentional signal modification by a third party, or severe signal degradation due to noise incurred along the signal path [10]. Vessel model inaccuracies cause the position estimate to diverge from the real vessel position over time. If nonlinear observers and IMU data for DR are used, the integration of velocity and acceleration measurements that contain errors cause the estimated position to diverge from the true position. Typical error sources of IMU sensors are bias, misalignment relative to the vessel frame axes and temperature variation [11].

In this paper we design a data-based method for DR that involves modelling the horizontal velocity of the vessel in terms of inputs like thruster command/feedback values, thruster power consumption, measurable environmental states, and heading. A Long Short-Term Memory (LSTM) neural network was used for this purpose due to its ability to handle large time delays between input data and the resulting effect

on the output data.

As the heading of the vessel may be measured by an internal sensor system such as the compass, availability of the heading measurement is independent of the GNSS system. Thus the change in position may be calculated from the heading angle, the predicted body-fixed velocity at the next time step, and the sampling time. Adding this to the position derived at the previous time step results in the estimated DR position at the next time step. Figure 1 shows a schematic view of the general units required. Under normal operation the "Velocity" unit provides target values for the supervised training. The target values have corresponding input vectors lagged by one sample step and consist of variables related to the actuators of the vessel, wind conditions and the vessel heading. The "Initial learning" block performs offline training based on the sampled targets and inputs. If a GNSS failure occurs, no targets are available, which precludes any further supervised training. At this point the input vector is used to form one-step predictions of the longitudinal (surge) and lateral (sway) velocity of the vessel. The proposed method

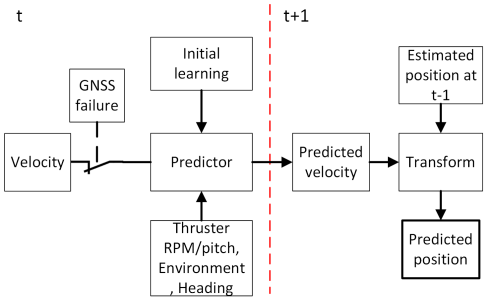


Fig. 1. The proposed approach for performing DR using machine learning methods.

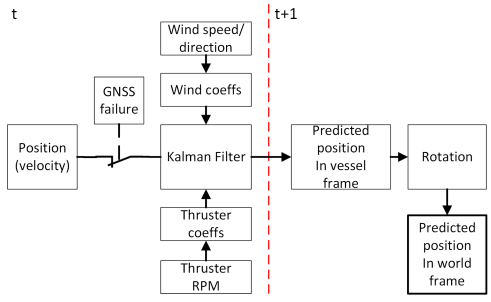


Fig. 2. The approach for performing DR using the KF.

has the advantage of not being dependent on a mathematical model of the vessel. Thus it offers a more generic way of representing the velocity/position of a vessel due to force input by thrusters and other relevant and obtainable measurements. In addition, automatic parameter adaptation can be performed purely based on sampled data. This may be relevant if, for example, the load distribution on the hull changes during operation. On the other hand, state estimators, such as the KF and nonlinear observers, allow for proof of stability, as well as a more transparent input/output relationship. Figure 2 shows how the KF approach to DR may be performed. For both figure 1 and figure 2 the vertical red line marks the line between measured position signals and predicted position signals. A comparison in terms of position estimation performance was made between the two methods to gauge the feasibility of the LSTM model for DR.

The remainder of this paper is organized as follows. Section II reviews related literature. Section III introduces the model used for predicting the vessel motion, how it is configured in terms of input, architecture and hyperparameters and also the signals generated by the vessel simulator. Results from two case studies, along with a description of the simulated vessel and the environmental disturbances imparted on it, are given in Section IV. Section V provides a discussion on the results from Section IV and Section VI offers a conclusion on the performance and validity of the proposed method.

II. Related work

The DR mode is a position reference fallback system for marine surface vessels. Vessels operating beneath the ocean surface may apply DR positioning techniques as the primary system of determining position [12]. German et al. compared two methods of determining position for an Autonomous Underwater Vehicle [13]. Internal sensors included a three-axis magnetic compass, a Doppler Velocity Log and a depth sensor. The first method relied on an Extended Kalman Filter fusing Global Positioning System (GPS) data, transferred acoustically from an autonomous tender vessel, with the onboard sampled data. The second used only the internal sensors, which produced dead-reckoned position solutions.

For DR of ocean surface vessels, Diamant and Jin used a three-axis accelerometer to provide the dead-reckoned heading and position of a vessel [14]. They used machine learning to classify accelerometer data into bins of similar pitch angle and then project it onto the local north-east horizontal plane. The projected accelerations were integrated to yield the estimated position and heading. The motivation for using only a three-axis accelerometer as sensor input for DR was to avoid using measurements from a gyrocompass. According to the authors this sensor may be unavailable or contain too much noise to be of use in estimating the attitude of the vessel.

Rogne et al. investigated the DR capabilities of an INS aided by dGNSS signals [9]. They applied two different low-cost IMUs, providing accelerometer, compass, and angular velocity measurements. Two different nonlinear observers were compared, using no information about the vessel model, on a test set sampled on a vessel performing a DP operation in the North Sea. They found that the top performer had a position error, after 10 minutes of dGNSS outage, of about 100 m.

DR has been used in other domains as well, such as the automobile and aerospace. When comparing seagoing vessels with airplanes, it is clear that there is a large difference in

dynamic properties and how severe the impact of wind is on the frame of the respective objects. This is especially true for Unmanned Aerial Vehicles (UAVs) due to their small size. Mokhtarzadeh and Gebre-Egziabher performed a study on cooperative navigation for UAVs [15]. Several UAVs, connected in a network, shared navigational information during a 5 minute GPS outage to reduce the position error drift rate of a DR based navigation filter. The authors opted to use an integration of airspeed measurements, instead of the more traditional INS sensors in order to avoid the double integration necessary to determine position from the acceleration estimated by the INS. An additional advantage to this approach is the separation of the DR operation from the Attitude and Heading Reference System. Instead of using an airspeed sensor Fusini, Johansen, and Fossen used a downward-looking camera and a machine vision system to provide the velocity of the UAV [16]. The acquired velocity was input to both a nonlinear observer and an exogenous KF for performing DR, in which a bounded error rate was achieved during experimental real-system testing.

Land vehicles usually follow predefined tracks, often in areas that are not conducive for robust GNSS signal reception. To produce continuous in-car navigation services, DR/INS systems, digital maps and mathematical models of the vehicle typically complement the GNSS measurements. Skog and Händel provide an overview of such systems, and the methods used for fusing both external sensor data (e.g. GNSS) and internal sensor data (e.g. odometer, gyroscopes, and accelerometers) [17].

Abbott and Powell provided a study of the error contribution of various sensors for an in-car navigation system [18]. They applied sensitivity analysis to gauge the performance of a KF sensor fusion algorithm against a reference system. Their findings suggested that the use of differential GPS (dGPS) offered improved calibration of the internal sensors, resulting in significant reduction of error drift during a satellite system outage. Thereby, relatively inexpensive internal sensors combined with dGPS could provide sufficiently accurate DR systems. Extending the flexibility of the KF for combining data from several sensors at various sampling rates, Barrios, Motai, and Huston introduced a dynamic state noise covariance matrix [19]. The purpose of this dynamic matrix is to reflect the state uncertainty more accurately when sensors drop out for any length of time.

Like Rogne et al. [9], Ahmed and Tahir [20] recognize that high-performance IMU units contribute significantly to the overall system cost. That motivated the use of a low-cost Micro Electro-Mechanical System IMU unit, containing a triaxial gyroscope and accelerometer, to accurately determine the attitude of a car. They estimated the vehicle acceleration by using the kinematic vehicle model and the known norm of the gravity. In addition to providing accurate attitude estimates, the ability to separate the gravity-induced acceleration components from the overall acceleration measurement proved beneficial to DR performance.

III. Methodology

In this section we introduce the measured signals, delays present in the actuators of the simulated vessel and the LSTM network model. Methods of limiting the input data dimension and selection of LSTM hyperparameters are also considered.

A. Measurement noise

Noise was added to the following measured states.

- Position: The position measurements given in the North East Down (NED) frame.
- Heading: Rotation about the z-axis of the vessel.
- Velocity: The linear velocity given in the NED frame.

The position and heading measurements, as seen by the consumers of the sensor data, are, then, a sum of the true value sampled from the simulator, white noise, a bias, and a Gauss-Markov (GM) process. Equation 1 shows the discretized GM process:

$$x[k+1] = \exp\left(-\frac{\Delta t}{T_c}\right) x[k] + \sigma w[k] \tag{1}$$

where k is the discrete time variable, Δt is the sampling interval, T_c is the correlation time, and w is the Gaussian white noise with a standard deviation of σ. Equations 2 and 3 show the addition of noise terms to form the expression for the position and heading with noise [21]:

$$\boldsymbol{p}[k] = \boldsymbol{p}_{true}[k] + \boldsymbol{x}_p[k] + \boldsymbol{\sigma}_p \boldsymbol{w}_1[k] + \boldsymbol{\mu}_p \tag{2}$$

$\boldsymbol{p}$ is a two-dimensional column vector containing the north and east position with additive noise, $\boldsymbol{p}_{true}$ is the noiseless north/east position, $\boldsymbol{x}_p$ holds the corresponding GM processes for the two components, $\boldsymbol{\sigma}_p$ is a diagonal matrix containing standard deviations of added white noise ($\boldsymbol{w}_1$) and $\boldsymbol{\mu}_p$ holds the position bias.

$$\psi[k] = \psi_{true}[k] + x_\psi[k] + \sigma_\psi w_2[k] + \mu_\psi \tag{3}$$

Noise added to the heading signal is described in Equation 3, where ψ is the heading angle containing noise, ψ_{true} is the noiseless heading angle, x_ψ is the GM process related to the heading angle, σ_ψ is the standard deviation of the Gaussian white noise w_2 and μ_ψ is the heading angle bias. Table I shows the parameters used in simulating the position and heading states with noise. The angular/linear velocity received only a constant bias and white noise [22].

TABLE I
PARAMETERS USED FOR THE ADDITIVE NOISE ELEMENTS OF THE POSITION AND HEADING MEASUREMENTS.

	GM		White noise	Bias
	σ	T_c	σ	μ
Position	0.1 m	240 s	0.2 m	[-0.2,0.2] m
Heading	0.1 °	60 s	0.1 °	[-0.1,0.1] °

B. Time delay

Delays in time between a change in thruster command (input) and the given response in velocity (output) are present in the sampled time series. They are caused by both the linear/rotational inertia of the vessel and the rotational inertia of the various thruster systems. Figure 3 shows the surge velocity response due to a step increase of 5 degrees in the commanded pitch angle of the two main thrusters. At a pitch angle of 5 degrees the thrusters output about 5 % of the maximum thrust force. We see that the rate of change of the thruster itself is limited to 1.4 degrees per second, such that it takes approximately 3.5 s to reach 5 degrees. Furthermore, the time spent to reach a surge velocity of 63 % of the steady state value of 0.37 $\mathrm{m\,s^{-1}}$ is 50 s.

During normal DP operation there will be no step function inputs as the controller reaches a relatively fixed command vector to compensate for the external disturbances. However, perturbations in thruster commands occur due to imperfect wave filtering, causing setpoint changes in the range $[-0.5, 0.5]$ degrees. To ensure that the input vector to the machine learning algorithms contain information of the most significant transient effects, due to changes in thruster commands, we include 10 seconds of history data for each input variable. Similar delays are seen for the tunnel thrusters. Delays also exist between the vessel velocity and the changes in wind velocity and direction.

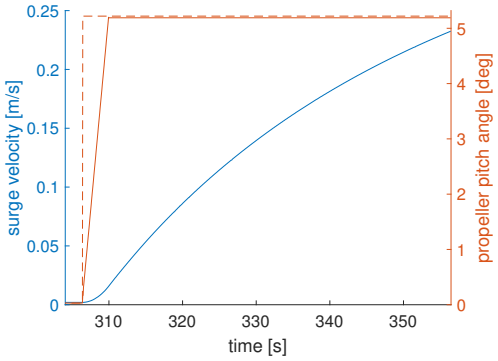


Fig. 3. The delayed response of the two variables surge velocity (solid blue) and actual thruster pitch angle (solid red) as a reaction to a step increase in the commanded thruster pitch angle (dashed red).

C. Position estimation concept

Two networks predict the horizontal velocity components of the vessel: one that predicts the surge velocity and one that predicts the sway velocity. This makes it possible to provide a custom network in terms of input pattern for each of the velocities expressed relative to the horizontal axes of the vessel frame of reference. After an initial network learning phase the proposed approach does not rely on samples produced by a GNSS system. Inputs to the networks are therefore available up to, and including, the discrete step k. This enables a prediction of the velocities at the subsequent time step, $k+1$. To get from a predicted velocity to a predicted travelled distance in the NED frame, the predicted velocity is multiplied by the sampling time and rotated according to the heading angle. At this point the travelled distance due to the predicted velocity, Δp in Figure 4, is added to the previously estimated position. Equation 4 gives the equation for the propagation of position

$$\hat{\boldsymbol{p}}[k+1] = \hat{\boldsymbol{p}}[k] + \boldsymbol{R}(\psi)\hat{\boldsymbol{v}}[k+1]\Delta t \tag{4}$$

where $\hat{\boldsymbol{p}}$ is the estimated north/east position of the vessel in the NED frame, $\hat{\boldsymbol{v}}$ is the predicted velocity vector relative to the vessel frame coordinate system, and $\boldsymbol{R}(\psi)$ is the square rotation matrix that transforms the predicted velocities to NED-frame velocities. $\hat{\boldsymbol{v}}$ contains the surge and sway velocity of the vessel, variables ($\hat{v}_{lon}, \hat{v}_{lat}$ of Figure 4). k is the discrete step index with a step interval of $\Delta t = t[k] - t[k-1]$. A visualization of the process is given in Figure 4. At time $t[k]$ the horizontal position is measured using the signal received from GNSS satellites. At the next time step, $t[k+1]$, the receiver on the vessel fails to produce the position of the vessel via GNSS signals due to one of the aforementioned reasons for GNSS unavailability. At this point, the DR algorithm is activated and provides an estimate of the vessel position through the prediction of the surge ($\hat{v}_{lon}$) and sway ($\hat{v}_{lat}$) velocities seen in Figure 4. Together they make up the velocity vector $\hat{\boldsymbol{v}}[k+1]$ of Equation 4. The

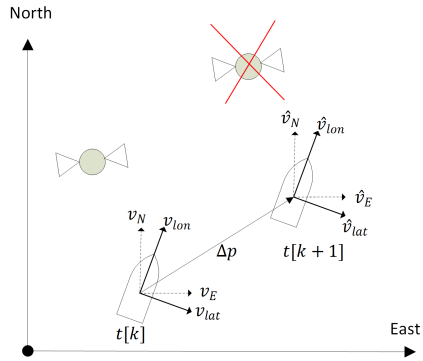


Fig. 4. A switch from normal operation (t[k]) to loss of GNSS system, requiring a DR system to estimate the position at the next step without an absolute position measurement.

method proposed in this paper, the LSTM recurrent neural network, by design only receives input variables that contain information about external disturbances, the heading angle, and the control intention of the vessel. Measurable external disturbances include the wind velocity and wind direction for the system used in this paper. Although systems exist for measuring and estimating the wave spectrum parameters in the vicinity of the vessel [23], [24] and measuring the velocity and direction of the ocean current affecting the hull [25], we limit the environmental sensory equipment to sensors that are currently available in the system. A key assumption at this stage is that the velocities relative to the vessel frame are

available without bias. If the velocity targets used for training the machine learning methods contain biases, the error rates during DR are increased significantly.

D. LSTM

A LSTM network was used to model how the velocity of the vessel relates to the aforementioned inputs. LSTM networks differ from feedforward networks in that they have weight connections between all nodes that are not input nodes [26]. To avoid the problem of vanishing/exploding gradients for backpropagation-through-time learning, Hochreiter and Schmidhuber devised a unit called a memory cell [27]. It contains a Constant Error Carousel (CEC) unit that aims to keep the error flow constant through a unity self-connection. A linear activation is used in the CEC. The memory cell contains two multiplicative gate units in addition to the CEC. They control the access of the input signals and output signals to the CEC. As LSTM networks are particularly well suited for learning the relationship between events that are separated by a long time delay, we include this network in our analysis. Due to the large inertia of both the vessel and the various actuator systems, there may be delays between such events as the inputting of a command and significant position change. See Section III-B for a visualization of the time lag. Functions in the Matlab Neural Network toolbox were used for training and prediction using the LSTM network.

As sensors output measurements of various physical quantities, they operate in different value ranges. In order to have each measured variable contribute equally as part of the input vector, all data should be normalized. In order to scale both the variation and the absolute value of each variable in the dataset, we use the mean/standard deviation approach to normalization according to Equation 5

$$\boldsymbol{x}' = (\boldsymbol{x} - \bar{\boldsymbol{x}})/\text{std}(\boldsymbol{x}) \tag{5}$$

where $\boldsymbol{x}$ is the N-sample by M-variable training dataset, $\bar{\boldsymbol{x}}$ is the mean value of each variable, std() represents the standard deviation of the variables and $\boldsymbol{x}'$ is the normalized data. All of the signals used in this paper have a bounded range, meaning that given a representative set of training data, the range of the test data does not differ significantly.

E. Input selection

By limiting the number of input variables to those that hold a certain level of information about the output states, the network's ability to generalize increases and its complexity is reduced. Mutual Information (MI) is applied in this paper to facilitate the dimension reduction of the input vectors used by the machine learning models. This operation is known as input selection and is performed prior to generating, or updating, the actual predictive network. MI provides a measure of the reduction of uncertainty about a variable x given a variable y [28]. It is defined by

$$I[\boldsymbol{x}, \boldsymbol{y}] = -\int\int p(\boldsymbol{x}, \boldsymbol{y}) \ln\left(\frac{p(\boldsymbol{x})p(\boldsymbol{y})}{p(\boldsymbol{x}, \boldsymbol{y})}\right) dxdy \tag{6}$$

where $p(\boldsymbol{x})$ and $p(\boldsymbol{y})$ are the distributions of $\boldsymbol{x}$ and $\boldsymbol{y}$, respectively, and $p(\boldsymbol{x}, \boldsymbol{y})$ is the joint distribution between the two sets. Thus, if the evaluation of $I[\boldsymbol{x}, \boldsymbol{y_1}]$ results in a larger numerical value compared to the evaluation of $I[\boldsymbol{x}, \boldsymbol{y_2}]$, the variable $\boldsymbol{y_1}$ contains more information than the variable $\boldsymbol{y_2}$ about the variable $\boldsymbol{x}$. Estimators are employed for practical implementations of MI and its use within the domain of time series regression is documented in [29] and [30]. In this paper we calculate MI using the Matlab functions presented in [31].

1) Input structure: The vessel has six thrusters: two bow tunnel thrusters, two stern tunnel thrusters, and two main thrusters with rudders. In this paper, the vessel performs stationkeeping using one Proportional Integral Derivative (PID) regulator per Degree of Freedom (DOF), preceding a basic thrust allocation unit that applies the unconstrained generalized inverse method for distributing motion controller force requests. To simplify the allocation problem, the rudder angle of the two main thrusters was fixed. A further simplification was performed to decouple the effect of the main thrusters on the rotation of the vessel. For all simulations in this paper the main thrusters were operated in unison, such that they only affected the motion of the vessel along its longitudinal axis. By intuition we select inputs to represent the velocity of the vessel in its forward and sideways axes, individually. The forward/surge speed varies depending on the inertia, thruster force, and environmental force applied along that axis. Thus, measurements of the main thrusters (fixed along the forward axis) are included along with the wind direction and velocity and heading angle. Without a mathematical model of the effect of the thruster commands and wind magnitude and direction, we aim to derive this from the measurements. We take a similar approach in selecting the input variables for the velocity in the sway direction, selecting measurements from both a forward- and a stern-mounted thruster as well as the heading and wind measurements. Equation 7 shows the partitioning of the variables in an input pattern,

$$\begin{aligned}\boldsymbol{z}_k =& [x_1[k] + x_1[k-d] + \ldots + x_1[k-(n-1)d], \ldots \\ & x_2[k] + x_2[k-d] + \ldots + x_2[k-(n-1)d], \ldots \\ & x_m[k] + x_m[k-d] + \ldots + x_m[k-(n-1)d]]\end{aligned} \tag{7}$$

where $\boldsymbol{z}$ marks the total, one-dimensional, input pattern, k is the discrete sample step, x is the measured input variable, d is the delay in number of steps, n is the number of delayed samples to include of a variable and m indicates the type of input variable. See the first column of Table III for a list of input variables used in the two separate input patterns, which corresponds to the variable m.

2) Optimizing network structure: Depending on parameters such as neuron number, layer depth, size of training dataset, etc., the evaluation of a single instantiated neural network may be quite costly in terms of computation time. The approach of Snoek, Larochelle, and Adams, termed Bayesian optimization, provides efficient hyperparameter optimization, thereby lowering the overall cost of producing an efficient model configuration [32]. In this paper the optimization of the LSTM network (see Section III-D) was focused around the number of LSTM blocks in a single layer, as well

TABLE II
THE DIMENSIONS OF THE SIMULATED VESSEL.

Description	Value
L_{pp}	82.7 m
Breadth	23.1 m
Displacement	10180×10^3 kg

as the learning rate, the two most important parameters according to [33]. A range of [10, 200] was selected for the number of blocks while a range of $[10^{-5}, 10^{-1}]$ was selected for the learning rate. The cost function returned the mean-squared-error (MSE) of the validation samples (10% of the total number of samples used for training), which provided a means of quantifying the expected performance of the network. Together with the input selection stage, the number of parameters in need of tuning has now been limited to that of setting the threshold for the input selection and the upper/lower values of the range in which to perform hyperparameter optimization.

IV. SIMULATION RESULTS

We propose to use a data-based model, described in Section III-D, to model the relationship between various inputs and the predicted linear, vessel-frame relative, velocities of the vessel at the next time step. To assess the performance and the validity of this method, we compare it to two other models:

- KF: Linearized equations of motion are obtained for the vessel by rotating the position measurements to a vessel-parallel coordinate system at each time step. This facilitates the use of a linear KF observer model for the DP test case in this paper [34].
- SLFN: A single-layer feedforward neural network, which represents the most basic structure among neural networks used for regression.

In the case of the KF we coast through the outage using the thruster command, wind velocity, and wind angle as input. These measurements are fed to the mathematical model of the vessel. The individually learned predictive models of the two machine learning methods replace the explicit vessel model. The machine learning DR methods do not use the vessel model or sensors for measuring the displacement of the vessel.

A. Vessel and environment description

All experiments were conducted in a commercial simulator developed by the Norwegian company Offshore Simulator Centre AS. It features a simulated environment in which a user may manipulate the wind, waves, and ocean current to mimic real-life conditions. It offers a library of virtual vessels to choose from. For these experiments, a multi-purpose offshore vessel was selected. Table II provides its main dimensions. Figure 5 shows a view of the simulated environment with the selected vessel engaged in a DP operation close to a static rig. For the specific simulation study performed in this paper, varying environmental parameters were applied. The direction of the environmental disturbances is incremented at intervals of 30 degrees from 0 to 360 degrees, relative to the vessel frame. At each fixed direction a set of wind and wave magnitudes were applied consecutively, causing increasingly severe weather conditions. Table IV shows the wind and wave magnitude for each of the distinct conditions faced by the vessel at the directions previously specified. A specific weather condition is determined by the direction of the wind and waves along with their respective magnitudes. In this test set each weather condition has a duration of 14 minutes, of which the first seven minutes involves a change of both wind and wave magnitude from the previous weather condition. If all conditions have been run for a single direction this transition period involves a linear transition from one weather direction to the next one. The entire simulation test set spans approximately 15 hours of vessel maneuvering. The actual run time is reduced by means of running the simulation 5 times faster than the real time.

A three DOF DP controller is applied to perform station-keeping. The controller applies a single PID controller in each DOF and the output of the motion controller connects to a basic generalized inverse control allocator for distribution of the generalized force vector into individual thruster commands. Figure 9 shows how the true position compares to the position with measurement noise added (see Section III-A). The latter is the raw position output by the dGNSS system when it is operating normally. The noiseless position signal is not used for any other purpose than visualization.

1) KF parameters: A KF was implemented for comparison to a conventional method of DR. It requires model-dependent matrices in addition to tuning parameters. We list the applied tuning parameters along with the matrices describing the mass and damping of the simulated vessel in the following paragraphs.

$$M = \begin{bmatrix} 1.02e7 & 0 & 0 \\ 0 & 1.02e7 & 8.44e6 \\ 0 & 8.44e6 & 5.80e9 \end{bmatrix} \tag{8}$$

$$D = \begin{bmatrix} 300000 & 0 & 0 \\ 0 & 550000 & 600000 \\ 0 & 600000 & 1.38e8 \end{bmatrix} \tag{9}$$

Furthermore, the two tuneable matrices of the KF, the $\boldsymbol{R}$ and $\boldsymbol{Q}$ matrices had the following numerical values. Note that the values in $\boldsymbol{R}$ were determined using a dataset sampled while the vessel was unaffected by environmental disturbances, while the general rules given in [6] were used for tuning the $\boldsymbol{Q}$ matrix.

$$\boldsymbol{R} = \text{diag}([0.7, 0.7, 0.2]) \tag{10}$$

$$\begin{aligned} \boldsymbol{q}_1 &= \text{diag}([0.1, 0.1, 0.1]) \\ \boldsymbol{q}_2 &= \text{diag}([1e6, 1e6, 1e6]) \\ \boldsymbol{q}_3 &= 0.1 \times \boldsymbol{R} \\ \boldsymbol{Q} &= \begin{bmatrix} \boldsymbol{q}_1 & \boldsymbol{0}_{3x3} & \boldsymbol{0}_{3x3} \\ \boldsymbol{0}_{3x3} & \boldsymbol{q}_2 & \boldsymbol{0}_{3x3} \\ \boldsymbol{0}_{3x3} & \boldsymbol{0}_{3x3} & \boldsymbol{q}_3 \end{bmatrix} \end{aligned} \tag{11}$$

In terms of objective, the implementation of the KF used in this paper differs from the other methods. The KF aims

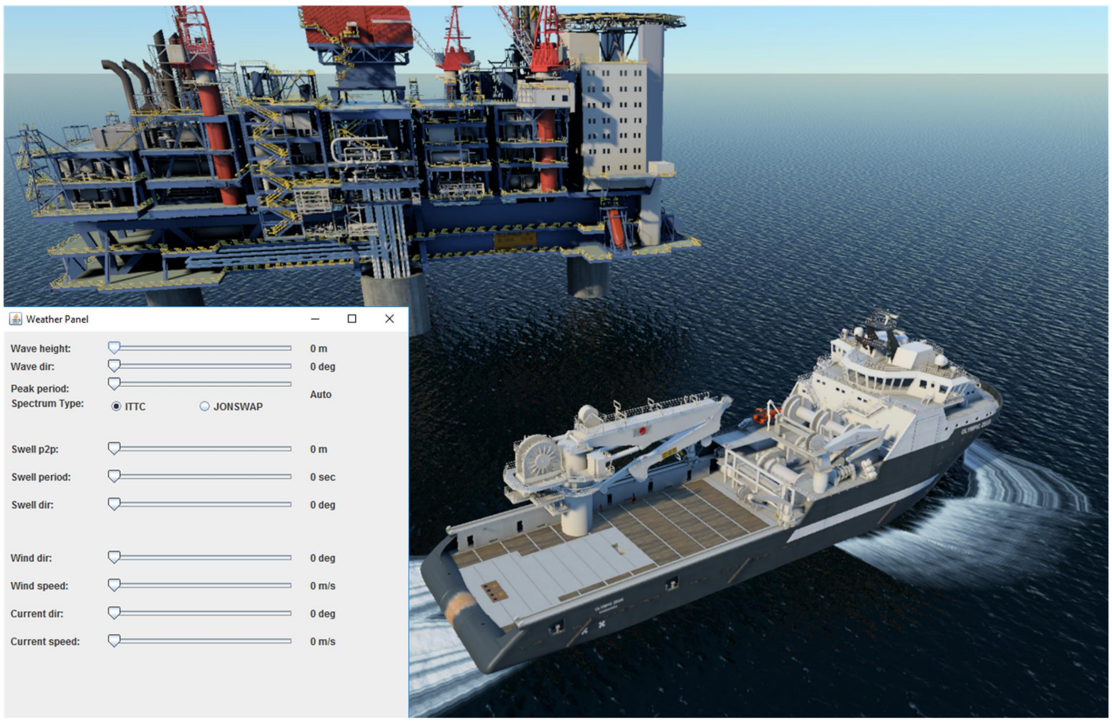


Fig. 5. A screenshot that shows the simulated environment and the panel for applying environmental disturbances.

to produce a position estimate that reflects no influence caused by zero-mean oscillatory wave forces. Thus, under normal operation we would expect to see a smooth trajectory following the mean of the measured position and heading. The other two methods aim to copy the exact behaviour of each time series, resulting in a more erratic trajectory during normal operation due to both measurement noise and wave-induced motion.

B. Case study 1: input selection

Reducing the input dimension of the network has positive effects on computation time as well as network interpretability and generalization ability. It is key to retain sections of the overall input pattern that contain useful information, which sets the stage for the method described in Section III-E: Mutual Information. MI allows for a ranking of input variable importance relative to an output variable. Therefore, input variables that offer a low relative MI value was deselected at this stage. In Table III we see the 0-1-normalized MI of the two target variables; surge velocity and sway velocity.

As shown in the "Description" column of Table III, *power* indicates the consumed power in watts of the specific thruster, *cmd* indicates the command sent to the thruster (either a blade pitch angle or an angular velocity value), and *act* indicates the feedback value measured at the thruster. Given the results in Table III and a threshold value of 0.4 we see that the reduced input pattern of the network predicting the surge velocity consists of input variables 2, 3, 10, and 12. For the network predicting the sway velocity the variables are 2, 3, 4, and 7. The input patterns are thereby reduced to 66 % (surge velocity) and 44 % (sway velocity) of the original input length. The dataset used for training contains 10^4 samples spaced by one second. Over the course of about 2.5 hours of simulation time, 12 randomly chosen weather conditions are run. Wave heights and wind velocities were chosen within the ranges given in Table IV.

TABLE III
THE NORMALIZED AVERAGE MI VALUE OF INPUT VARIABLES RELATIVE TO THE OUTPUT VARIABLES.

Input variable	Description	Surge velocity	Sway velocity
1	Heading angle	0.0	0.08
2	Wind angle	1.00	0.69
3	Wind velocity	0.67	0.69
4	Bow thruster power	-	1.00
5	Bow thruster cmd	-	0.02
6	Bow thruster act	-	0.04
7	Stern thruster power	-	0.73
8	Stern thruster cmd	-	0.00
9	Stern thruster act	-	0.01
10	Main thruster power	0.49	-
11	Main thruster cmd	0.37	-
12	Main thruster act	0.44	-

A comparison of the performance in terms of estimated

position, relative to the sampled true position, is seen in Figure 6. It displays the mean error with/without MI over a one minute DR period for all weather conditions in the test set where the vessel was able to keep the desired position. The deselected weather conditions are highlighted in Section IV-C. As noted in Section IV-A, each individual weather condition lasts for 14 minutes, of which one minute towards the end of each weather condition was applied for the DR tests. Using the complete input vector for both the surge velocity estimator and the sway velocity estimator results in an increase in position error. Figures 7 and 8 show the result

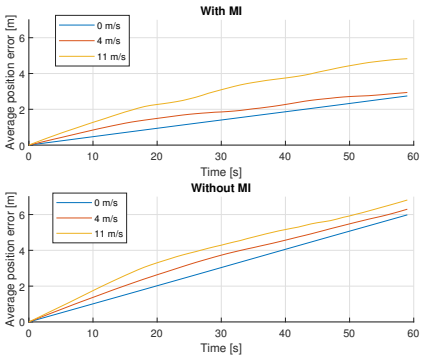


Fig. 6. The mean DR error of the LSTM method for all wind directions at wind velocities of 0, 4 and 11 $\mathrm{m\,s^{-1}}$. Top plot: a subset of the elements in the complete input vector was extracted using MI and used as input to the LSTM model. Bottom plot: all entries in the complete input vector was used as input to the LSTM model.

of running the optimization function to determine optimized hyperparameters for the LSTM estimators. For the reduced-input estimators of surge velocity and sway velocity the following hyperparameter pairs were selected based on the lowest observed MSE value:

- Surge velocity: block number = 43, learning rate = 0.0070
- Sway velocity: block number = 26, learning rate = 0.0165

C. Case study 2: impact of the environmental variables

In this section we look at how the LSTM, SLFN, and KF perform over a wide operational range. According to the previous section, input selection is applied, resulting in the use of variables (2, 3, 10, 12) to predict the future surge velocity and variables (2, 3, 4, 7) to predict the future sway velocity (see Table III). Figure 9 shows the position of the vessel throughout the test set, with and without measurement noise. Similar to the previous section we use the position without measurement noise as reference. To evaluate the models we view the mean distance error observed during a one minute period of each weather condition. Each period of evaluation, in which DR is required, starts three minutes after the transition into the new weather condition has finished. This allows the control algorithm time to adapt to the current

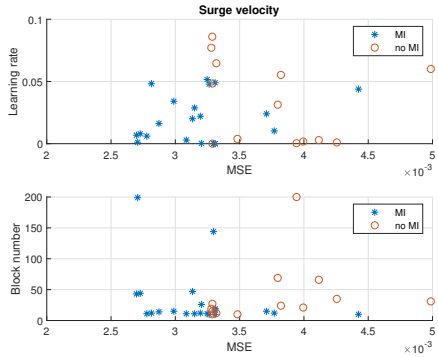


Fig. 7. Results in terms of running the LSTM hyperparameter optimization function on the two parameters learning rate and block number for the surge velocity estimation model.

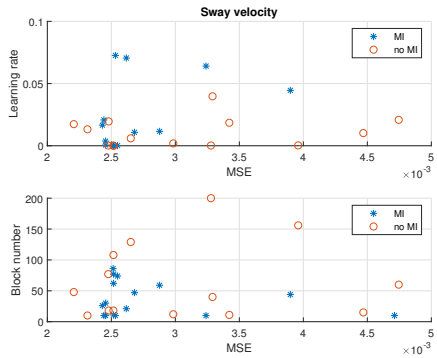


Fig. 8. Results in terms of running the LSTM hyperparameter optimization function on the two parameters learning rate and block number for the sway velocity estimation model.

environmental forces. Figure 10a shows how the distance error propagates, without any GNSS input, for the LSTM approach. The distance from the origin of the figure to each discrete weather direction is determined by Equation 12, which gives the mean position estimation error.

$$\bar{e}_{dist}[k] = \frac{1}{N}\sum_{k=1}^{N}\sqrt{(\hat{p}_n[k]-p_n[k])^2+(\hat{p}_e[k]-p_e[k])^2} \tag{12}$$

TABLE IV
THE PARAMETERS OF THE SEA STATES SIMULATED AT EACH DISCRETE WEATHER DIRECTION.

Significant wave height (Hs)	Wind velocity
1 m	2 $\mathrm{m\,s^{-1}}$
2 m	4 $\mathrm{m\,s^{-1}}$
3 m	7 $\mathrm{m\,s^{-1}}$
4 m	11 $\mathrm{m\,s^{-1}}$

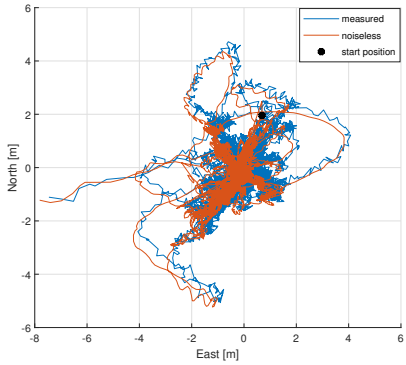


Fig. 9. Visualization of the vessel position for a part of case study 2. The noiseless position measurement (red line) is only included to provide a reference to the raw dGNSS position output (blue line).

(p_n, p_e) is the measured horizontal plane position given in the NED frame, $(\hat{p}_n, \hat{p}_e)$ is the corresponding dead-reckoned position, and k signifies the discrete step. Figures 10b and 10c show the results of running the same test with the remaining two methods, i.e., the SLFN and the KF, respectively.

Considering the area covered by the polygons in Figure 10a, the mean error is roughly similar irrespective of weather direction and wave heights below three meters. Similar properties are seen for the SLFN method (see Figure 10b). The optimization scheme used for the LSTM method was also applied for the SLFN method. This yielded an optimized hidden neuron number of 93 for the sway velocity estimator and 55 for the surge velocity estimator. For both estimators, the optimization procedure favoured a sigmoid activation function. An increased DR error may be seen for both methods at weather directions of 270, 120, 90, and 60 degrees. At these directions and a wave height of four meters the tunnel thrusters are unable to produce sufficient thrust to withstand the environmental forces acting on the vessel. This caused saturation of thruster commands and divergence from the desired position. When a given set of thruster commands no longer cause vessel motion similar to that experienced in the training set (e.g. when the environmental forces outweigh the control forces and cause thruster saturation) the output of the estimators diverge from the true vessel velocity. The most severe effects of the saturation are seen at a direction of 120 degrees and 4 m wave height. The vessel is unable to recover the desired position in a timely fashion, causing further estimation error for all simulated conditions at the subsequent weather direction of 90 degrees.

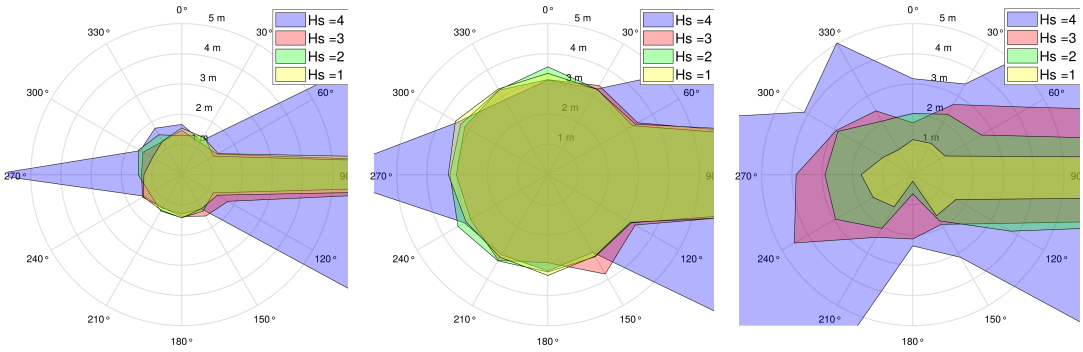
V. Discussion

The input variables related to thruster command, thruster operating point, and power do not directly give information about the motion of the vessel. However, they indirectly contain information about how the vessel moves. A thruster command, executed over a given time interval, induces forces on the vessel, causing a change in linear/angular speed. The consumed power fluctuates both due to the thruster command and the velocity of the vessel relative to the surrounding water. Accounting for lags (see Section III-B), one may obtain knowledge of how the vessel moves by viewing thruster data. This is one of the advantages of using a data-based model: it learns such connections. To make the task of the machine learning methods easier, and make them more effective, input selection picks the most relevant input variables. Input selection also mitigates the issue of the curse of dimensionality for our problem, which is an issue for high-dimensional input patterns in regression problems [35]. The number of samples necessary to approximate a function to a certain degree of smoothness grows exponentially with the input dimension.

In this study we performed input selection on the basis of the mean MI (see Section III-E) for an input variable containing lags according to, for example, variable x_1 of Equation 7. This allows for an uninterrupted representation of the selected variable. Another strategy would be to select the entries of the total input pattern (see Section III-E1) that has an MI value greater than some threshold, which does not leave the inter-variable spacing intact, but ensures that all entries in the selected pattern have a given MI content relative to the target variable.

The results produced in Section IV-B show the increased performance gained by selecting input variables that provide a certain amount of information about the output variable, omitting the remainder of the original input variables. Viewing the optimization results in Figures 7 and 8, we see that only the surge velocity estimator benefits from applying MI, at least in terms of the MSE derived from a validation set consisting of 10 % of the samples in the training dataset. This amounts to roughly 1000 samples. Although the sway velocity estimator displays a slight decrease in performance when applying the reduced input pattern, the overall effect of MI is positive. As the input selection process of the two estimators are separate, one may choose to implement one, or both, of the reduced input patterns in order to maximize the expected DR performance. Figure 6 displays how the estimated position, using input vectors selected by MI, diverges more slowly compared to applying the original input pattern during a GNSS dropout. As MI was shown to aid the LSTM model (see Section III-D) in terms of reducing the position estimation error, it was applied to both machine learning models for the second case study, shown in Figure IV-C. Of the two, the LSTM performed best with a mean distance error of less than 2 m for wave heights below 3 m. The measurements of thruster-related states (power consumption, setpoint, and feedback) were assumed to be noise-free.

The KF, described in Section IV-A1, has similar performance relative to the LSTM for wave heights of 1 m. When wave heights of 2, 3, and 4 m affect the vessel, the LSTM provides consistent DR position estimates while the KF error increases. The KF error increase is, in part, due to the linear relationship between a thruster command and the resulting force output of a thruster assumed in a regular KF. Due to the lack of measurements to facilitate a corrector-function, the DR position is driven solely by the vessel model and the



(a) LSTM (b) SLFN (c) KF

Fig. 10. Results of mean position estimation error given in the horizontal plane for the LSTM, SLFN and KF model for case study 2. Each data point shows the mean position estimation error during a one-minute DR period. Hs denotes the significant wave height in meters.

thruster command input vector. Similar to the LSTM model the SLFN model displays consistent DR position estimates, although at a larger magnitude. When wave heights exceed 3 m, the SLFN outperforms the KF. As the implemented KF requires a significant number of parameters to be set, an optimization scheme to derive optimized KF parameters might offer a more balanced comparison between the three methods.

While the KF requires no initialization process, it does require a mathematical model of the vessel. Machine learning models create an equivalent model based on data. That is why, from a cold start, the machine learning algorithms require a certain amount of time to construct and train the estimator. During this time the DR functionality is unavailable. While this is inconvenient, it may be remedied by performing the initialization process at regular intervals, or continuously, in order to have a DR model that is current with respect to the state of the vessel. Thereby, it can seamlessly provide position estimates to a vessel operator, or the underlying automatic control system of a vessel, during a position reference system outage. This requires either maintaining a window of the most recent samples to perform batch training or feeding each individual sample to an online training algorithm for each of the two machine learning methods. This is particularly important as we make the assumption of constant mean environmental forces during the DR process. If an "old" model is used, it may not reflect the characteristics of the current environmental state. In our approach the training set consisted of 12 random weather conditions, which was assumed to be representative of the complete set of possible weather conditions. The authors acknowledge that the relative performance of the methods proposed in this study is highly affected by the value of the parameters of each method.

VI. Conclusion

Through the simulation studies we have compared the proposed LSTM NN method with a conventional KF and a SLFN model. When no position or velocity measurements are available, the three methods utilize their own established model together with their related model inputs. For the KF these are given by the vessel model as thruster commands and forces due to wind. But for the machine learning methods, an initial input pattern was selected, then input selection reduced this to a vector comprised of about two-thirds of the most relevant entries of the original input vector. This offered an improvement in terms of position estimation performance.

Findings suggest that the models created by machine learning methods offer comparable performance in terms of position error drift, without requiring any vessel-dependent parameters. This shows that the dynamics of the vessel may be modelled without the development and tuning of a mathematical model. However, machine learning methods offer no guarantees of convergence, being inherently black-box. Therefore future research should provide a deeper investigation into how to establish a measure of confidence into the behaviour of the machine learning methods.

Acknowledgement

This work was supported in part by the project SFI MOVE and Digital Twins for Vessel Life Cycle Service (Project no.: 237929 and 280703). The third author was partially funded by the Norwegian Research Council (NTNU AMOS) at the Norwegian University of Science and Technology (grant no. 223254). The authors would like to thank the staff of the Offshore Simulator Centre AS for their assistance related to interfacing with their simulator.

References

[1] M. Laurinen, "Remote and Autonomous Ships: The next steps," Advanced Autonomous Waterborne Applications (AAWA) Initiative, Tech. Rep., 2016. [Online]. Available: http://www.rolls-royce.com/{~}/media/Files/R/Rolls-Royce/documents/customers/marine/ship-intel/aawa-whitepaper-210616.pdf

[2] J. Bhatti and T. E. Humphreys, "Hostile Control of Ships via False GPS Signals: Demonstration and Detection," *Navigation, Journal of the Institute of Navigation*, vol. 64, no. 1, pp. 51–66, 2017.

[3] A. J. Sørensen, "A survey of dynamic positioning control systems," *Annual Reviews in Control*, vol. 35, no. 1, pp. 123–136, 2011.

[4] M. Carter, "Moving Towards a Standardized Interface for Acoustic Inertial Reference Systems," in *Proceedings of Dynamic Positioning Conference*, 2014.

[5] O. Cadet, "Introduction to Kalman Filter and its use in Dynamic Positioning Systems," in *Proceedings of Dynamic Positioning Conference*, 2003.

[6] T. I. Fossen and T. Perez, "Kalman Filtering for Positioning and Heading Control of Ships and Offshore Rigs: Estimating the effects of waves, wind, and current," *IEEE Control Systems*, vol. 29, no. 6, pp. 32–46, 2009.

[7] J. G. Balchen, N. A. Jenssen, E. Mathisen, and S. Sælid, "A dynamic positioning system based on Kalman filtering and optimal control," *Modeling, Identification and control*, vol. 1, no. 3, pp. 135–163, 1980.

[8] T. H. Bryne, T. I. Fossen, and T. A. Johansen, "Design of Inertial Navigation Systems for Marine Craft with AdaptiveWave Filtering aided by Triple-Redundant Sensor Packages," *International Journal of Adaptive Control and Signal Processing*, pp. 1–23, 2015.

[9] R. H. Rogne, T. H. Bryne, T. I. Fossen, and T. A. Johansen, "MEMS-based Inertial Navigation on Dynamically Positioned Ships: Dead Reckoning," *IFAC-PapersOnLine*, vol. 49, no. 23, pp. 139–146, 2016. [Online]. Available: http://dx.doi.org/10.1016/j.ifacol.2016.10.334http://linkinghub.elsevier.com/retrieve/pii/S2405896316319218

[10] D. Russell, "Integrating GNSS and INS to Provide Reliable Positioning," in *Proceedings of Dynamic Positioning Conference*, 2012.

[11] B. Vik, A. Shiriaev, and T. I. Fossen, "Nonlinear observer design for integration of {DGPS} and {INS}," in *New Directions in Nonlinear Observer Design*, 1999, vol. 244, pp. 135–159.

[12] Z. Chu, D. Zhu, B. Sun, J. Nie, and D. Xue, "Design of a dead reckoning based motion control system for small autonomous underwater vehicle," in *2015 IEEE 28th Canadian Conference on Electrical and Computer Engineering (CCECE)*, 2015, pp. 728–733.

[13] C. R. German, M. V. Jakuba, J. C. Kinsey, J. Partan, S. Suman, A. Belani, and D. R. Yoerger, "A long term vision for long-range ship-free deep ocean operations: Persistent presence through coordination of Autonomous Surface Vehicles and Autonomous Underwater Vehicles," in *2012 IEEE/OES Autonomous Underwater Vehicles*, 2012.

[14] R. Diamant and Y. Jin, "A machine learning approach for dead-reckoning navigation at sea using a single accelerometer," *IEEE Journal of Oceanic Engineering*, vol. 39, no. 4, pp. 672–684, 2014.

[15] H. Mokhtarzadeh and D. Gebre-Egziabher, "Performance of networked dead reckoning navigation system," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 52, no. 5, pp. 2539–2553, 2016.

[16] L. Fusini, T. A. Johansen, and T. I. Fossen, "Dead Reckoning of a Fixed-Wing UAV with Inertial Navigation Aided by Optical Flow," in *2017 International Conference on Unmanned Aircraft Systems*, 2017.

[17] I. Skog and P. Händel, "In-Car Positioning and Navigation TechnologiesA Survey," *IEEE Transactions on Intelligent Transportation Systems*, vol. 10, no. 1, pp. 4–21, 2009.

[18] E. Abbott and D. Powell, "Land-vehicle navigation using GPS," *Proceedings of the IEEE*, vol. 87, no. 1, pp. 145–162, 1999.

[19] C. Barrios, Y. Motai, and D. Huston, "Intelligent Forecasting Using Dead Reckoning with Dynamic Errors," *IEEE Transactions on Industrial Informatics*, vol. 12, no. 6, pp. 2217–2227, 2016.

[20] H. Ahmed and M. Tahir, "Accurate Attitude Estimation of a Moving Land Vehicle Using Low-Cost MEMS IMU Sensors," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 7, pp. 1723–1739, 2017.

[21] T. H. Bryne, "Nonlinear Observer Design for Aided Inertial Navigation of Ships," Ph.D. dissertation, Norwegian University of Science and Technology, 2017.

[22] R. Mahony, T. Hamel, P. Morin, and E. Malis, "Nonlinear complementary filters on the special linear group," *IEEE Transactions on Automatic Control*, vol. 53, no. 5, pp. 1557–1573, 2008.

[23] L. P. Perera and C. G. Soares, "Weather routing and safe ship handling in the future of shipping," *Ocean Engineering*, vol. 130, no. February 2016, pp. 684–695, 2017. [Online]. Available: http://dx.doi.org/10.1016/j.oceaneng.2016.09.007

[24] U. D. Nielsen and J. J. Jensen, "A novel approach for navigational guidance of ships using onboard monitoring systems," *Ocean Engineering*, vol. 38, no. 2-3, pp. 444–455, 2011. [Online]. Available: http://dx.doi.org/10.1016/j.oceaneng.2010.11.024

[25] E. M. Bitner-Gregersen, S. K. Bhattacharya, I. K. Chatjigeorgiou, I. Eames, K. Ellermann, K. Ewans, G. Hermanski, M. C. Johnson, N. Ma, C. Maisondieu, A. Nilva, I. Rychlik, and T. Waseda, "Recent developments of ocean environmental description with focus on uncertainties," *Ocean Engineering*, vol. 86, pp. 26–46, 2014.

[26] J. Schmidhuber, "Deep Learning in neural networks: An overview," *Neural Networks*, vol. 61, pp. 85–117, 2015. [Online]. Available: http://dx.doi.org/10.1016/j.neunet.2014.09.003

[27] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[28] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2006.

[29] A. Sorjamaa, J. Hao, N. Reyhani, Y. Ji, and A. Lendasse, "Methodology for long-term prediction of time series," *Neurocomputing*, vol. 70, no. 16-18, pp. 2861–2869, 2007.

[30] T. Trappenberg, J. Ouyang, and A. Back, "Input variable selection: Mutual information and Linear Mixing Measures," *IEEE Transactions on Knowledge and Data Engineering*, vol. 18, no. 1, pp. 37–46, 2006.

[31] T. Suzuki, M. Sugiyama, T. Kanamori, and J. Sese, "Mutual information estimation reveals global associations between stimuli and biological processes," *BMC Bioinformatics*, vol. 10, no. SUPPL. 1, pp. 1–19, 2009.

[32] J. Snoek, H. Larochelle, and R. Adams, "Practical Bayesian Optimization of Machine Learning Algorithms." *Nips*, pp. 1–9, 2012. [Online]. Available: https://papers.nips.cc/paper/4522-practical-bayesian-optimization-of-machine-learning-algorithms.pdf

[33] K. Greff, R. K. Srivastava, J. Koutník, B. R. Steunebrink, and J. Schmidhuber, "LSTM: A Search Space Odyssey Klaus," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 28, no. 10, pp. 2222–2232, 2017. [Online]. Available: http://arxiv.org/abs/1503.04069

[34] T. I. Fossen, *Handbook of Marine Craft Hydrodynamics and Motion Control*. John Wiley & Sons Ltd., 2011.

[35] M. Verleysen and D. François, "The Curse of Dimensionality in Data Mining," in *IWANN 2005*, vol. 3512. Springer, Berlin, Heidelberg, 2005, pp. 758 – 770.

Robert Skulstad, Mechatronics Lab, Department of Ocean Operations and Civil Engineering, Norwegian University of Science and Technology (NTNU) Aalesund. E-mail: robert.skulstad@ntnu.no.

Guoyuan Li, Mechatronics Lab, Department of Ocean Operations and Civil Engineering, NTNU Aalesund. E-mail: guoyuan.li@ntnu.no.

Thor I. Fossen, Department of Engineering Cybernetics, NTNU Trondheim. E-mail: thor.fossen@ntnu.no.

Bjørnar Vik, Kongsberg Maritime, Aalesund. E-mail: bjornar.vik@km.kongsberg.com.

Houxiang Zhang, Mechatronics Lab, Department of Ocean Operations and Civil Engineering, NTNU Aalesund. E-mail: hozh@ntnu.no.

C

Paper III

A Hybrid Approach to Motion Prediction for Ship Docking— Integration of a Neural Network Model into the Ship Dynamic Model

Robert Skulstad, Guoyuan Li, *Senior Member, IEEE,* Thor I. Fossen, *Fellow, IEEE,* Bjørnar Vik, and Houxiang Zhang, *Senior Member, IEEE*

***Abstract*—While automatic controllers are frequently used during transit operations and low-speed maneuvering of ships, ship operators typically perform docking maneuvers. This task is more or less challenging depending on factors such as local environment disturbances, number of nearby vessels, and the speed of the ship as it docks. This paper proposes a tool for onboard support that offers position predictions based on an integration of a supervised machine learning (ML) model of the ship into the ship dynamic model. The ML model is applied as a compensator of the unmodelled behaviour or inaccuracies from the dynamic model. The dynamic model increases the amount of predetermined knowledge about how the vessel is likely to move and thus reduces the black-box factor typically experienced in purely data-driven predictors. A prediction horizon of 30 seconds ahead of real time during docking operations is examined. History data from the 29-meter coastal displacement ship RV (Research Vessel) Gunnerus is applied to validate the approach. Results show that the inclusion of the data-based ML model significantly improves the prediction accuracy.**

***Index Terms*—Ship motion prediction, supervised deep learning, onboard support**

I. Introduction

SHIP motion prediction is a general term that incorporates many elements. These include the states in which to perform predictions – for example prediction of ship orientation, position, or up/down motion – the temporal aspect (long, medium and short) and the model that makes the prediction of the states in the near future. Typically these predictions, which are based on time-series data, coincide with a specific application that could benefit from having information about future states of the vessel motion. Historically, research efforts have been focused on ship orientation and applications where safety or efficiency can be increased using predictions of these states. Mainly this is due to the abundance of operations that are severely impacted by angular motions of a vessel, including takeoff and landing of autonomous aerial vehicles and helicopters [1], crane operations [2] [3], and missile launch [4]. These operations can be made safer and more efficient by incorporating knowledge about future vessel states.

Docking is the task of maneuvering the vessel to a fixed mooring location. On the path towards the dock the vessel operator must tackle challenges such as passing/nearby vessels, compensating for forces induced on the vessel by environment disturbances, and arriving at the dock location in a timely fashion. The latter is especially important for ferries or vessels transporting goods on a fixed route, where keeping the time schedule is key. Although much effort has been put into ship autonomy in recent years [5], docking is still a largely manual task performed by the vessel operator. Research in the field of ship motion prediction typically focuses on methods within one domain, e.g. dynamic- or kinematic models and Kalman filters, machine learning (ML), deep learning or auto-regressive (AR) methods (see Section II).

Dynamic models aim at describing the motion of the vessel due to forces estimated by simplified representations of the vessel, including thruster effects and to some extent, forces due to environmental disturbances. Simplifications are necessary due to the lack of direct measurements of wave/current drift. Additionally, for docking applications, effects due to local wind fields, cushioning effects at the dock and shallow water exist, which are not measured directly. The true model is complex and nonlinear; thus a simplified model is often used and discrepancies between the behaviour of the real ship and the dynamic model are expected. Kinematic prediction models allow for translating motion measurements, such as accelerations into predictions of position. However, they account for neither the effects of thruster commands nor the direct effect of wind forces. While many ML methods are well suited for representing nonlinear models they require a substantial amount of sampled data to do so reliably. In addition, the inner connections in a ML model may not be readily understandable.

Examples where existing knowledge of the behaviour of

Robert Skulstad, Guoyuan Li and Houxiang Zhang are with the Department of Ocean Operations and Civil Engineering, Norwegian University of Science and Technology (NTNU), Aalesund, Norway

Thor I. Fossen is with the Department of Engineering Cybernetics, NTNU, Trondheim, Norway

Bjørnar Vik is with Kongsberg Maritime, Aalesund, Norway

the ship is utilized in cooperation with data-based ML models are scarce. This study will therefore investigate the feasibility of one such approach: making position predictions using a dynamic model while in parallel, an ML model predicts the position prediction error made by the dynamic model in order to compensate for any unmodelled behaviour or inaccuracies. Including the measured wind velocity and direction as input to the ML model, contributes significantly to the success of the proposed approach. While a Kalman filter could simulate the dynamic model and derive predictions in a similar manner, the ability to provide corrections to the dynamic model predictions, gained by learning from docking examples differing in port location, sea state and wind conditions, would be lost. This work will focus on the prediction of position during the docking operation of a regular displacement ship. Figure 1 shows a picture of the ship. Currently, this is a manual task, relying on the ship operator to make appropriate and timely corrections to actuators in order to safely dock the vessel. During this operation the ship operator must make many choices due to changing environment factors, regulations calling for proper interaction with nearby/crossing vessels, and the effects of applied actuator commands. To aid in making these choices this paper proposes an onboard support tool, which will provide the vessel operator with predictions of the vessel position. These predictions originate from the hybrid predictor and span 30 seconds into the future [6], hereafter termed the prediction horizon. Key contributions of this paper includes the construction of a hybrid model for prediction of the future motion of a ship, and the use of data sampled onboard a coastal ship for training of the data-based model as well as verification of the prediction performance.

The remainder of this paper is organized as follows. Section II presents previous research in this domain. Section III introduces the predictors and their architecture/parameters, Section IV gives results and describes the vessel and data selected for training and testing of the predictors. Section V presents the conclusions.

II. Related work

Research on ship motion prediction generally revolves around a model that processes time series data, where each input channel contains data sampled at a fixed time interval. Several metrics exist in which time series prediction models may be categorized. If the model uses existing explicit knowledge of how the vessel moves due to forces and/or velocities (dynamic/kinematic models), the term model-based predictors may be appropriate. If only sampled data is used to learn the behaviour of the vessel, the predictor is termed data-based. We may also distinguish methods based on if they can represent nonlinear behaviour. The subsections below outline the description and classification of existing methods for ship motion prediction.

A. Model-based motion prediction

This section introduces predictors applying predetermined knowledge of how the vessel behaves when maneuvering.

1) Dynamic model: Triantafyllou et al. used a standard Kalman Filter (KF) to estimate and predict the motion states of the two decoupled motion groups heave-pitch and roll-sway-yaw [7]. They found that in order for the KF to be successful, an accurate model of both vessel and sea state spectrum was required. For the latter requirement the estimation of the modal frequency of the spectrum was key to the performance of the KF.

Sutulo et al. [8] aimed at creating maneuvering models (dynamic or kinematic) that could be inexpensive to evaluate, and thus be used in tasks related to prediction and onboard support. According to the authors this could make applications such as model-based collision avoidance and onboard decision support for deciding control commands feasible due to the computational efficiency of the models.

2) Motion density functions: Instead of using the equations of motions to model the dynamic behaviour of vessel states, as Triantafyllou et al. had done, Sidar and Doolin constructed the linear KF using approximations of density functions of measured heave and pitch motions [9]. The density functions were obtained experimentally. This led to a KF of significantly lower dimension compared to the work of Triantafyllou et al. Measured heave and pitch time series were assumed to be stationary, narrow band, and stochastic for the duration of the prediction interval. The choice of a KF as a tool for making predictions using the motion density functions was motivated by its ability to produce predictions in real time.

The approach to ship motion prediction taken by Nielsen et al. also relies on density functions of time series data [10]. By deriving the observed autocorrelation matrix for variables largely dictated by the induced wave force, predictions of 15-60 seconds were made on a model-scale ship.

3) Kinematic model: Perera et al. proposed to use the Extended Kalman Filter (EKF) to estimate the translational motion states and predict the trajectory of a vessel by means of a curvilinear motion model (CMM) [11], [12]. States included in this model were heading angle, normal and tangential accelerations, forward (surge) speed, and sideways (sway) speed. By combining the EKF and the CMM the authors found that the estimated velocities and accelerations, which were estimated based only on noisy position measurements, converged quickly (within 15 seconds) to small variations around the true values. For the validity of the prediction they assumed constant accelerations, which is a strong assumption given the nature of vessel motion. This was acknowledged by the authors, deeming the approach valid only for short-term predictions.

Perera later modified his approach to use a vector dot and cross product algorithm for the prediction of vessel motion

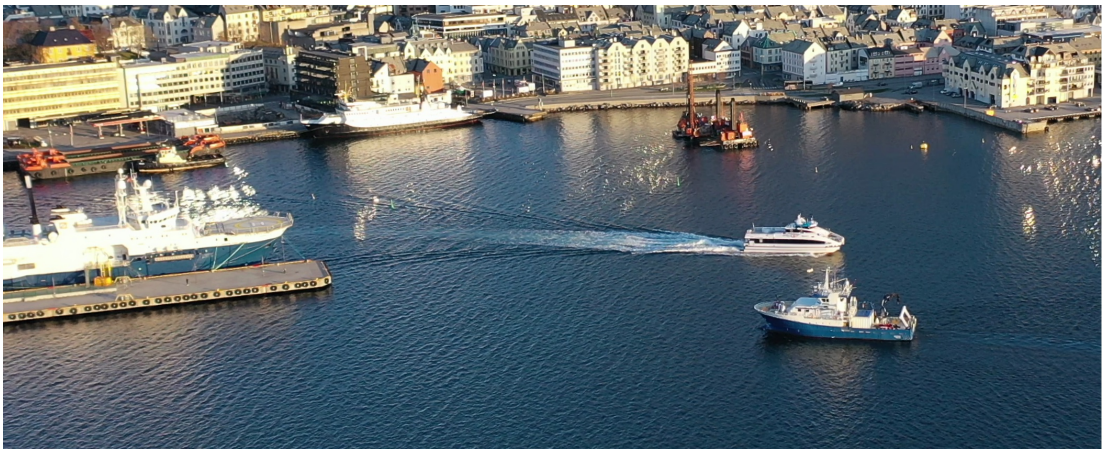


Fig. 1. The Research Vessel Gunnerus of the Norwegian University of Science and Technology (NTNU) (bottom right vessel) approaching a dock in the port of Aalesund, Norway.

[6]. Given the large inertia of a vessel, its trajectory creates a curve in the ocean plane, motivating the use of the CMM as vessel model in the EKF. Based on this property the algorithm calculates radii of the center of gravity and pivot point of the vessel relative to a calculated center of planar motion for a vessel. The states and parameters related to the vessel pivot point were estimated for a given time instance and used to predict the position and heading 30 seconds ahead of real time. The adaptability to varying conditions is, according to the author, preserved by the use of the EKF and predictions are valid under the assumption of constant navigation conditions within a short future time interval.

B. Data-based motion prediction

To address the difficulty of obtaining a sufficiently detailed mathematical model, which transitions the relevant states from one sampling time instance to the next, many researchers have turned to data-based predictor models. The principal advantage of such methods is the ability to construct a model that relates a certain sampled input vector to a certain output state vector without knowledge of the parameters of the physical object. This output state vector is a set of vessel states for which one wishes to determine numerical values ahead of real time. Support Vector Machines (SVM), neural networks, and AR models are examples of such methods and the majority of data-based predictor models used for ship motion prediction are varieties of these general models.

SVM features attributes such as strong generalization ability and global optimization [13]. Creating a model that is able to generalize well to inputs, beyond those provided in the learning stage of the method, is one of the key advantages of this method. Li et al. used SVM, together with several aiding methods, to predict the heave motion given waves impacting the vessel at four different directions [14].

The attributes of neural networks include the ability to adapt to input changes and to represent the nonlinear behaviour of the input-output relation of physical systems. Employing a time-delay neural network with wavelet activation functions and using sensitivity analysis to determine significant inputs, Zhang et al. performed prediction of the heading of a vessel a few steps ahead [15]. They concluded that this type of prediction may be used for the benefit of vessel control and safety.

Peng et al. applied data-based modelling to estimate the unknown ship dynamics as well as to reconstruct the unmeasured ship velocity. An Echo State Network [16] and a fuzzy system [17] comprise the tools that was integrated into an observer and subsequently used in vessel maneuvering control. The task of reconstructing the entire dynamic model of the vessel was relaxed through the introduction of a nominal mass matrix in [16]. Force produced by thrusters on the vessel was assumed to be known and subsequently input to the data-based model to approximate the vessel dynamics.

Zhang and Liu used a single layer feedforward network (SLFN) to predict the heading angle of a vessel a few sample intervals ahead [18]. This one-layer prediction network is common in the literature, although the choice of activation function, training method, number of hidden neurons, type and number of input variables and the number of input lags vary greatly. Arriving at suitable values for these parameters is the key challenge to providing reliable predictions using SLFNs. Often these parameters are derived using trial and error, although online pruning methods for producing compact

networks exist [19]. For statically trained networks genetic algorithms, grid search or random search algorithms may also be used to optimize architecture and hyperparameters.

Skulstad et al. applied a long-short term memory (LSTM) network, a version of a recursive network, to maintain estimates of position and heading of a ship during loss of position reference signals from the Global Navigation Satellite System [20]. They used a deep neural network similar to the one described in Section III-C. However, only a one-step prediction was made and sensors not relying on external signals, such as the compass, wind sensor, and sensors measuring operating conditions of the thrusters, were still active.

Maintaining accurate estimates of position and attitude during loss of GNSS signals is also of importance in the automotive and aerospace domain. Examples of approaches to mitigate such a sensor loss, through one-step predictions, are given in [21] and [22]. The former applies a KF in combination with an AR integrated moving average model and a feedforward neural network to predict errors accumulating in the inertial navigation system while the latter makes use of a radial basis function neural network for predicting the KF measurement update.

The AR method makes use of history samples of the target state accompanied by predictor model parameters determined by a least squares method [23]. It offers low computational costs, but has drawbacks handling nonlinear, non-stationary series [**?**]. Derivative methods to mitigate the effects of these drawbacks exist, such as Nonlinear AR method and Time varying AR method. To predict the displacement of a landing deck on a vessel, Yang improved upon the standard AR method by using Bayes Information Criterion to determine the number of model coefficients, and a forgetting factor to reduce the effect of older vessel states on the regression algorithm output [1].

Lately, studies on vessel traffic management have resulted in more emphasis on trajectory prediction in order to improve operational safety in congested waters [12] [24] [25]. However, these are longer-term predictions and fall outside the scope of this work. Similarly, prediction of a maximum envelope of roll, pitch, or heave motion, termed quiescent period prediction (see [26] and [27]) is out of scope for the present study.

Prediction using time series methods similar to the ones described above are also found in various other domains, such as: weather prediction [28], electrical load forecasting [29] and automotive motion prediction [30].

III. Predictor modelling

As the proposed method of this study is a combination of two predictor models originating from two separate fields; model-based and data-based, the following sections will outline how they are constructed and how they cooperate to predict the future ship motion.

A. Hybrid predictor

In order to utilize the two complementary predictors for creating a hybrid predictor, the vessel model predictor (see Section III-B) will act as a foundation, predicting the complete position state due to the sampled data it receives. As the relative water velocity is not measured onboard the vessel, nor are the effects of the waves on the vessel motion directly accounted for, a certain disagreement between the actual motion of the vessel and the position prediction output by the vessel model is expected. To compensate for the prediction errors made by the vessel model, the ML model is applied (see Section III-C). In this way, the hybrid prediction is the sum of the prediction made by the vessel model and the ML model. Figure 2 shows how the two models are combined to create a predictor of the future ship position. The top dashed box of Figure 2 shows the individual components of the vessel model. See Section III-B for a detailed description.

A fundamental difference between the vessel model and the ML model is the way they produce prediction output. While the ML model directly outputs predictions for the entire prediction horizon (for future times $t_h = [1 - 30]s$) the vessel model requires iterations. Thus, during training of the hybrid predictor, for each time instance in the input data the vessel model is iterated 30 times in order to produce targets for supervised training. This is illustrated by the block named *Actual ship position* and the subtraction of the position predicted by the vessel model, $\boldsymbol{\eta}$. During this iterative process, external signals such as thruster RPM and angle, wind speed and angle, and measured velocities are not updated as they are not known for future time instances. However, feedback loops are present inside the vessel model, causing dynamic behaviour within the prediction horizon in terms of thruster forces and vessel velocities. Training the ML predictor involves using the position error targets and the associated input vector to get optimized hyperparameters that reflect the dynamics of the error model (see Section III-C). As there are no feedback loops between the targets and the input vector of the ML supervised learning approach, it is termed open-loop. Description of the variables included in the input vector may be found in Table III. The bottom dashed box of Figure 2 is repeated, applying identical hyperparameter values, so as to create an ensemble of LSTM predictors (more on this in Section III-C1). To get hybrid position predictions during a docking operation, the sum of the vessel model position prediction, $\boldsymbol{\eta}$, and the LSTM model error prediction is calculated.

B. Vessel model

The vessel model uses established relations between actuators, external environmental disturbances (wind) and the

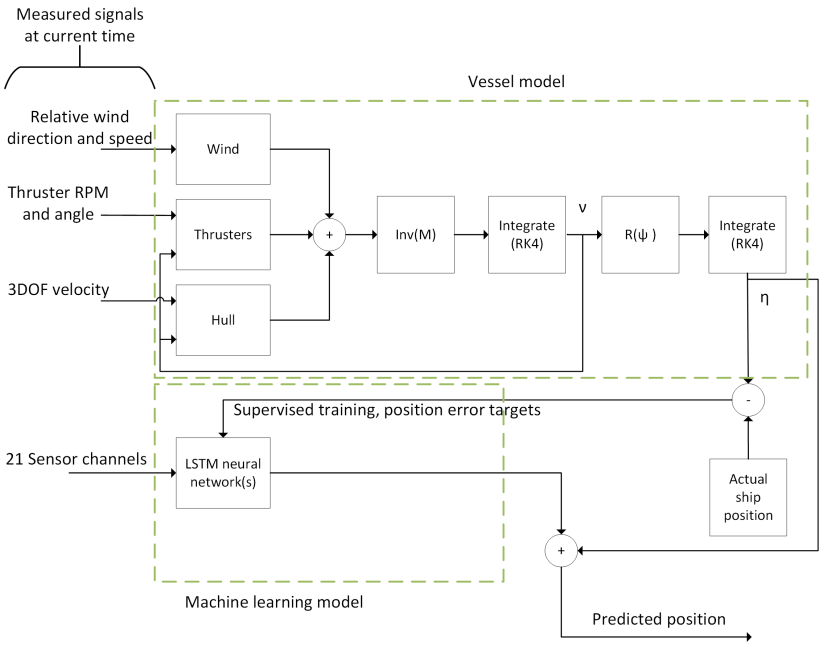


Fig. 2. The prediction strategy showing the vessel model predictor (top dashed green box) and the ML model (bottom dashed green box).

motion of the hull through water to describe the forces acting on the hull through the maneuvering model of Fossen [31]. The kinematic model is

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

where $\boldsymbol{\eta}$ is the pose vector containing the positions and yaw angle. $\boldsymbol{R}(\psi)$ is the horizontal plane rotation matrix due to the yaw angle, ψ. $\boldsymbol{\nu}$ is the velocity vector in surge, sway and yaw directions, respectively. Forces due to wind, waves, thrusters, hull friction, and inertia are given in (2)

$$\begin{aligned}&\boldsymbol{M}_{RB}\dot{\boldsymbol{\nu}} + \boldsymbol{C}_{RB}(\boldsymbol{\nu})\boldsymbol{\nu} + \boldsymbol{M}_A\dot{\boldsymbol{\nu}}_r + \boldsymbol{C}_A(\boldsymbol{\nu}_r)\boldsymbol{\nu}_r + \\ &\boldsymbol{D}\boldsymbol{\nu}_r + \boldsymbol{D}_n(\boldsymbol{\nu}_r)\boldsymbol{\nu}_r = \boldsymbol{\tau}_c + \boldsymbol{\tau}_{wi} + \boldsymbol{\tau}_{wa}\end{aligned} \tag{2}$$

where $\boldsymbol{\nu}_r = \boldsymbol{\nu} - \boldsymbol{\nu}_c$ is the relative ship velocity, and $\boldsymbol{\nu}_c = \begin{bmatrix} u_c & v_c & 0 \end{bmatrix}^T$ is the current velocity. $\boldsymbol{M}_{RB}$ is the rigid body mass matrix, $\boldsymbol{M}_A$ is the added mass matrix and $\boldsymbol{C}_A$ and $\boldsymbol{C}_{RB}$ are matrices describing the Coriolis/centripetal forces. $\boldsymbol{D}$ and $\boldsymbol{D}_n(\boldsymbol{\nu}_r)$ are linear and nonlinear damping matrices due to the hull moving through water. $\boldsymbol{\tau}_c$, $\boldsymbol{\tau}_{wi}$ and $\boldsymbol{\tau}_{wa}$ are forces on the ship due to thrusters, wind, and wave effects respectively. At this point a few simplifications to the vessel model become relevant:

- **Simplification 1:** Forces due to current are not accounted for in the vessel dynamic model (see (2)). The speed and direction of the current is not measured. Therefore the velocity of the ship relative to the water, represented by $\boldsymbol{\nu}_r$ in (2), is not known. $\boldsymbol{\nu}_r$ is therefore substituted by $\boldsymbol{\nu}$ in (2).
- **Simplification 2:** Forces due to waves, given as $\boldsymbol{\tau}_{wa}$ in (2), are not accounted for. This is due to the lack of measurements of the wave state. Besides, ports provide shelter from waves experienced in open ocean. Therefore we do not include estimates of forces from waves in the vessel model.

A numerical model of the forces produced by the two main azimuth thrusters was supplied by the thruster manufacturer. It is valid for all 4 quadrants of operation for the propeller (see Table I) and thus covers the key phases of the docking procedure of this study: the initial approach (transit), deceleration (windmilling) and low speed maneuvering. An introduction to this type of propeller model is given in [32]. With regards to the force produced by the bow tunnel thruster, only nominal force is estimated through a thruster curve provided by the thruster manufacturer.

To translate the propeller thrust into the three-dimensional force, $\boldsymbol{\tau}_c$, the azimuth angle and distance from the center of

TABLE I
THE 4 QUADRANTS OF PROPELLER OPERATION PARAMETERIZED BY RPM AND INFLOW VELOCITY (COURTESY OF [32]).

Parameter	**1**st	**2**nd	**3**rd	**4**th
n	≥ 0	< 0	< 0	≥ 0
V_a	≥ 0	≥ 0	< 0	< 0

gravity of the vessel to each thruster is applied (see (3)).

$$\boldsymbol{\tau}_c = \begin{bmatrix} 0 & c(\alpha_p) & c(\alpha_s) \\ 1 & s(\alpha_p) & s(\alpha_s) \\ l_{tx} & l_{px}s(\alpha_p) & l_{sx}s(\alpha_s) \\ & -l_{py}c(\alpha_p) & -l_{sy}c(\alpha_s) \end{bmatrix} \times \begin{bmatrix} T_{tn} \\ T_{pa} \\ T_{sa} \end{bmatrix} \tag{3}$$

l_{tx}, l_{px} and l_{sx} are the distances along the longitudinal axis of the vessel from the vessel center of gravity to the tunnel thruster, port main thruster and starboard main thruster respectively. α_p is the azimuth angle of the port main thruster while α_s is the azimuth angle of the starboard main thruster. The distance from the vessel center of gravity to each of the two main thrusters along the lateral axis of the vessel is given by l_{py} and l_{sy}. $s(\cdot)$ represents the sine function while $c(\cdot)$ represents the cosine function. Forces produced by each thruster along the propeller axis are given by the variables T_{tn}, T_{pa} and T_{sa} for the tunnel thruster, port main thruster, and starboard main thruster, respectively. Only lateral force and torque about the up-down axis of the vessel is produced by the bow tunnel thruster.

Wind force is the only external disturbance in which we use a deterministic model to estimate force. This is because the wind (velocity and direction) is the only one of the three environmental states measured. The three-dimensional force is given in (4).

$$\boldsymbol{\tau}_{wi} = \frac{1}{2}\rho_a V_{rw}^2 \begin{bmatrix} C_X(\gamma_{rw})A_{Fw} \\ C_Y(\gamma_{rw})A_{Lw} \\ C_N(\gamma_{rw})A_{Lw}L_{oa} \end{bmatrix} \tag{4}$$

where ρ_a is the density of air, V_{rw} is the relative wind velocity, γ_{rw} is the relative wind angle, C_X, C_Y and C_N are wind coefficients specific for the hull/superstructure shape. A_{Fw} and A_{Lw} are frontal and lateral projected areas and L_{oa} is the overall length of the ship.

C. Machine learning model

Several choices exist when selecting a method for the ML predictor. According to previous work in the domain of ship motion prediction using ML (see Section II-B), SVMs, neural networks (feedforward and recursive), and AR methods are popular choices. We will apply an LSTM network, which has shown outstanding performance in time-series modelling and prediction.

The sequential nature of time-series data related to motion of ships makes the LSTM a natural choice when searching for a representative model. This network type is specifically designed to store data over an extended period of time, allowing it to capture the relatively slow changes observed in data related to ship motion. Through the use of constant error flow, embodied by the Constant Error Carousels (CECs) in each LSTM block, and multiplicative gates that learn when to allow access to the CEC, events, or relations between input- and output data, spaced by a significant time interval, may be approximated [33]. In order to ensure satisfactory performance of the LSTM in predicting future vessel states, hyperparameters need to be set. This is done using the Matlab software, specifically the Bayesian optimization algorithm described in [34]. To limit the search space, and thus the required computation time, three parameters were included in the search:

- Learning rate
- Number of LSTM layers
- Number of blocks per layer

1) Ensembles: Due to randomness in the weight initialization of the LSTM network, each instantiation of a network with equal hyperparameters will output slightly different predictions faced with the same input data. By averaging the output of several networks, using the same optimized hyperparameters, the prediction error on previously unseen data can be reduced [35].

TABLE II
PHYSICAL PARAMETERS OF THE VESSEL USED IN THE EXPERIMENT.

Parameter	**Description**	**Value**
m	Mass of vessel	370 t
DWT	Deadweight	107 t
L_{pp}	Length between perpendiculars	28.9 m
Bm	Breadth middle (m)	9.6 m
dm	Draught (m)	2.7 m

IV. EXPERIMENT

Table II shows the main physical dimensions of the RV Gunnerus, a research vessel owned by the Norwegian University of Science and Technology. In terms of propulsors, two azimuth thrusters are mounted at the stern as well as a bow tunnel thruster. The two azimuth thrusters are each driven by a 500 kW electric motor, while the electric motor driving the bow thruster is rated at 200 kW. This yields a cruising speed of about 10 knots.

A. Data

The experiment was conducted based on history data acquired through log files created by a data acquisition system onboard the RV Gunnerus. A one-year time period was selected starting from August 2016 and ending in June 2017. For all variables in the data set a sampling rate of 1 Hz was observed.

In order to isolate successful dockings in the 2016-2017 period, three sensor channels were used. Two Boolean signals

TABLE III
THE VARIABLES USED IN THIS STUDY AS INPUT TO THE VESSEL MODEL AND LSTM MODEL.

Variable name	Unit	Range (train)	Range (test)
North	m	1688/-2703	2267/-769
East	m	1961/-486	1145/-1567
Heading angle	deg	360/0	360/0
Surge speed	knots	11.8/-1.82	11.98/-0.31
Sway speed	knots	1.43/-1.43	1.05/-1.03
Heading rate	deg/s	3.67/-3.43	2.84/-2.81
Roll	deg	2.68/-3.9	2.99/-2.72
Pitch	deg	0.53/-2.09	-0.02/-1.82
Heave	m	0.14/-0.12	0.35/-0.41
Roll rate	deg/s	2.22/-1.97	2.03/-2.17
Pitch rate	deg/s	1.08/-1.28	0.89/-0.69
Heave rate	m/s	0.17/-0.18	0.29/-0.25
Wind direction	deg	360/0	360/0
Wind speed	knots	19/0	15.6/0
Course	deg	360/0	360/0
Total speed	knots	11.8/0	12/0
Port thruster RPM	%	93.96/-67.64	99.19/-51.89
Port thruster angle	deg	121.93/-90.33	156.04/-89.83
Starboard thruster RPM	%	93.95/-67.58	100.08/-56.62
Starboard thruster angle	deg	106.33/-117.33	90.33/-146.44
Tunnel thruster RPM	%	102.1/-99.8	93/-61.2

originating from the propulsion system, *drive_running* (going from true to false) and *motor_at_zero_speed* (equals true), were applied in combination with a requirement of having a total speed of less than 0.1 m/s. When the docking time instances were successfully determined, 1000 samples prior to these instances were extracted and made up the data set for each docking operation. This interval may contain an initial period of automatic waypoint following control. However, the majority of the time is spent in the manual control mode, in which the ship operator guides the vessel to its docking location. Twenty-one sensor channels related to the motion of the vessel were sampled (see the first column of Table III), leading to a 1000x21 matrix of measurements per operation, spanning 15 locations along the west coast of Norway (see Figure 3).

Table III gives all the input variables for the hybrid predictor used in this study. Ranges are given as maximum and minimum values observed in the time series of each variable during 88 separate docking operations. Of these the first 68 were used for training and the last 20 were kept for testing purposes. The unit *deg* is short for degrees. To get a clearer sense of the nature of each variable, and the extreme values observed in the training data compared to the testing data, the max/min values are given in columns 3 and 4 of Table III.

A further processing of the position of the ship was made in order to generalize the position coordinates across docking locations. A conversion from position given as latitude and longitude in the earth-centered, earth-fixed (ECEF) frame to the local north-east-down (NED) frame in meters was

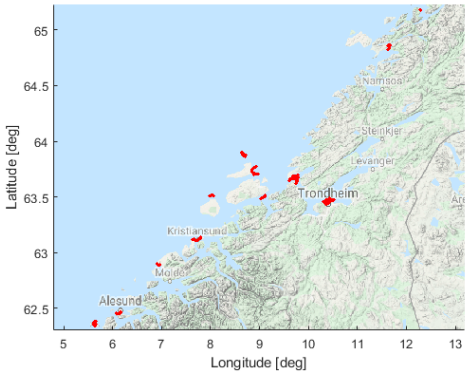


Fig. 3. The various docking locations of the RV Gunnerus along the west coast of Norway.

performed for convenience. The ECEF position recorded at the docking time instance was used as the origin for the NED coordinates of each docking operation. Figure 4a and 4b shows the path taken by the vessel towards the docking location at coordinates (0,0) m. The former shows all paths included as training instances for the ML algorithms, while the latter shows the test instances.

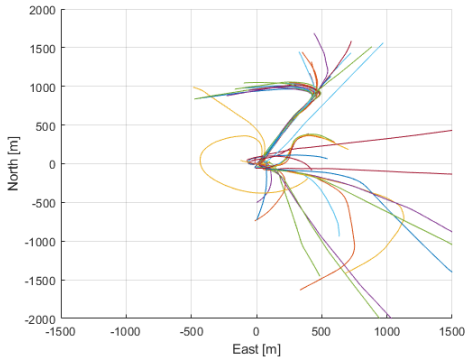
B. Prediction for one docking approach

As mentioned in the previous section, training data for the ML predictor consisted of 68 individual docking operations. First running the vessel model predictor on each time instance (1000 instances per docking operation), predicting 30 seconds ahead of real time, made it possible to generate an error signal by subtracting the vessel model position prediction from the actual position of the vessel. Thereby, the training targets, one vector with a 30-second prediction horizon per sampling instance, for the supervised training of the LSTM networks was created. Figure 5 shows the accuracy of the predictions in terms of average distance errors, calculated by (5), in the North-East plane for each docking operation in the training data.

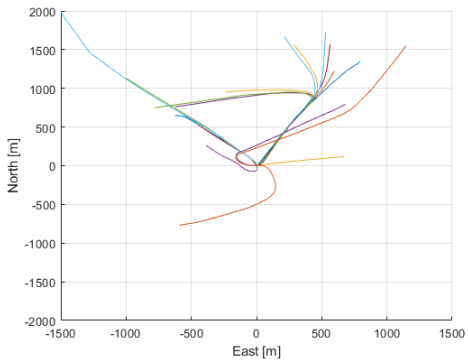
$$\bar{y}_{err,i} = \left(\sum_{j=1}^{M} \sqrt{(N_{ij} - \hat{N}_{ij})^2 + (E_{ij} - \hat{E}_{ij})^2} \right) / M \quad (5)$$

$\bar{y}_{err}$ is the mean distance error between the predicted and the true position of the ship in the prediction interval, M is the number of samples per docking operation and $i \in [1, 30]$ is the index of the prediction horizon, t_h. N and E represent the true north and east position, respectively, while $\hat{N}$ and $\hat{E}$ are the predicted north and east positions.

Given input data according to Section III-B at a certain time instance, the vessel model predictor iteratively predicts



(a) Training



(b) Testing

Fig. 4. The North-East path taken by the RV Gunnerus for all docking operations used in this study.

the position of the vessel 30 seconds ahead. In parallel, the ML predictor offers predictions of the position prediction error made by the vessel model predictor. The sum of these two 30-element vectors constitute the end result of the prediction approach: a set of coordinates given for future time instances in the North-East plane.

This is visualized in the plots of Figure 6, which shows the actual track of the docking approach and predictions made every 45 seconds. The red circle marks the start of each prediction, while the green star marks the end for both the hybrid predictor and the vessel model predictor. The red star shows the true position at the prediction interval end. If the red and green stars overlap, the position is predicted perfectly at 30 seconds ahead of real time. As the area between the true track of the vessel (red solid line) and the hybrid prediction vector (black dashed line) is smaller compared to the vessel model prediction vector, the hybrid

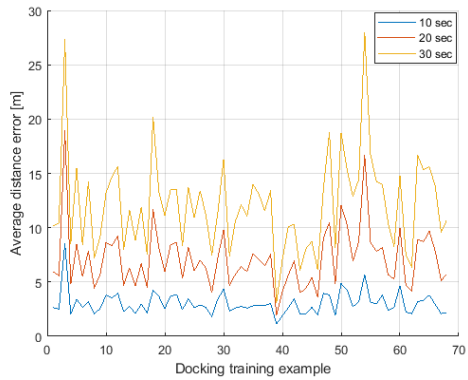


Fig. 5. The average distance between the true position and the position estimated by the vessel model at 10, 20, and 30 seconds prediction horizons on the training examples.

predictor has better position prediction accuracy. This is also evident from the prediction error for the prediction horizon end (distance between each green star and the adjacent red star).

Both the vessel model predictions and the hybrid model predictions diverge from the true position close to the origin. This is attributed to the use of thrusters to temporarily push the vessel against the dock while preparing the mooring ropes. Figures 7 and 8 show that while the speed of the vessel approaches zero at $t \approx 700s$, indicating that the vessel has docked, the thrusters are still producing thrust. In the same time period the course angle of Figure 7 is invalid due to zero speed. At $t \approx 630s$ the top plot of Figure 8 shows the port thruster being rotated. This is to push the vessel towards the dock with the starboard side facing the dock.

The top plot of Figure 7 plots the course angle against the heading angle. For the final approach to the Trondheim docking location, the course and heading angle deviate by several degrees in the time period 300-500 s. This is due to the dock being located in the outlet of the Nidelva river, and the water flowing towards the sea induces force on the hull. The effect on the vessel model prediction was a steady error of approximately 5 m in the North-East plane during this time period.

C. Average performance

Figure 9 shows the performance of the hybrid predictor and the vessel model predictor. The number of samples for the averaging of the position prediction at future times $t_h = [1, 30]s$ includes the entire 1000 seconds prior to completing the docking operation. Typically this involves a short initial period of transit speed, followed by deceleration to a

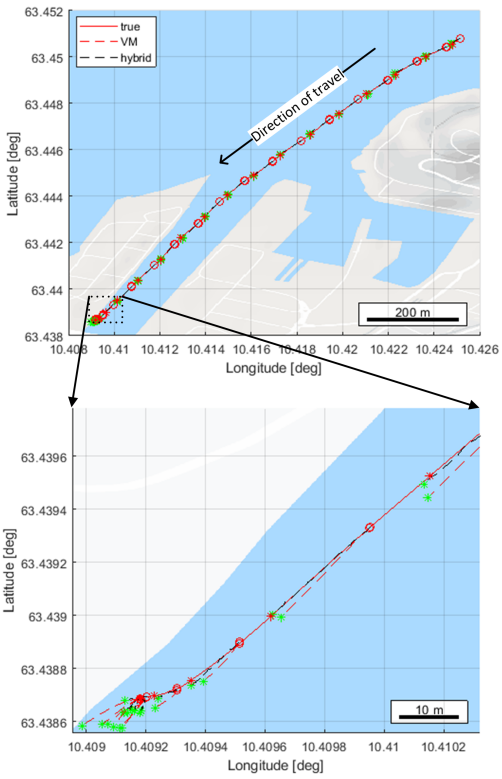


Fig. 6. The prediction of ship position in the horizontal plane in the port of Trondheim, Norway.

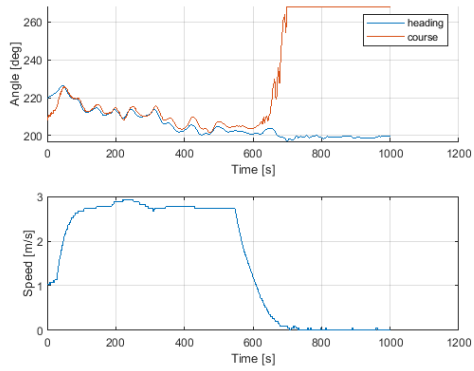


Fig. 7. The top plot shows the heading and course angle of the ship while docking in Trondheim, while the bottom plot holds the ship speed.

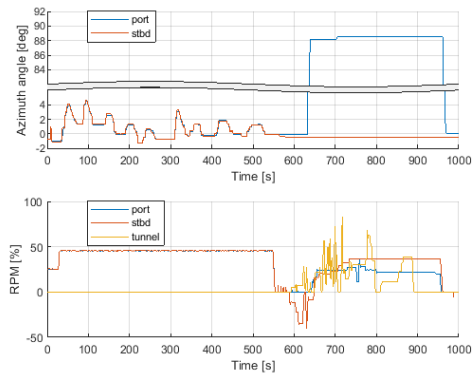


Fig. 8. The top plot shows the rotation angle of the main thrusters, while the bottom plot shows the RPM percentage of all three thrusters.

low-speed approach and finally gliding/decelerating towards close-to-zero speeds. This final period usually involves the application of the bow tunnel thruster (see the bottom plot of Figure 8). Both the vessel model predictor and the LSTM model predictor can introduce prediction error to the overall hybrid predictor. However, the main sources of error come from the vessel model and are due to the following reasons:

- the vessel model predictor does not account for forces induced by current as they are not measured;
- change in the control commands input by the vessel operator is unknown within each prediction interval and therefore the initial value is applied.

The LSTM predictor assumes a portion of this error, which results in an improved overall position prediction, reducing the average prediction error at $t_h = 30s$ by almost half. While there are three docking operations that exhibit close to the same accuracy as the vessel model prediction average (three black lines close to the red dashed line of Figure 9), two of them are at a docking location not covered in the training data set. The third line is generated by predictions carried out while docking in a port, which is covered only once in the training data set, and in an irregular fashion as well. It is irregular in the sense that the vessel did not follow the usual pattern of deceleration, but moved toward the dock in lurches. The remaining black lines in Figure 9 depict the prediction errors incurred while docking at more frequently visited docking locations.

Due to the nature of data-based models, where training data dictates the performance of the trained model, the more repetitions of docking at a certain port will lead to the hybrid predictor providing better predictions at this location. For the application described in this paper, if the trained

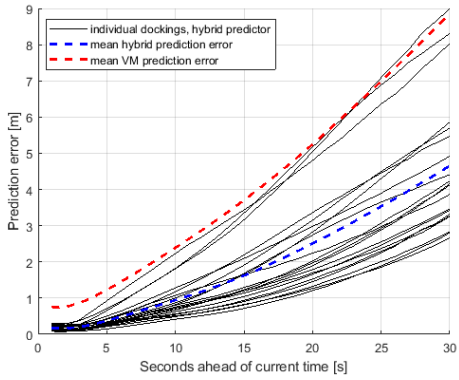


Fig. 9. The average position prediction error of the vessel model predictor by itself (dashed red line) and the hybrid predictor (dashed blue line) over the 20 test sets and the prediction horizon (1-30 seconds). The solid black lines represent the average position prediction error of the 20 individual docking operations included in the test set for the hybrid predictor.

hybrid predictor was applied while docking the ship at a port for the first time, it would perform worse than if it docked at its home port of Trondheim. However, by including the vessel model predictor, which provides a deterministic evaluation of the position prediction, along with facilities in the LSTM predictor to maintain its generalization abilities (regularization, early stopping training and dropout layer), an increase in position prediction accuracy is seen compared to the predictions made by the vessel model predictor alone. Downsides to this hybrid predictor include the requirement of having both sufficient amount of data for training of the data-based model as well as the parameters of the dynamic model described in Section III-B.

The second column of Table IV displays the averages of the values seen for $t_h = [10, 20, 30]s$ in Figure 5. A lower average prediction error is observed for the testing data relative to the training data. This is due to the diversity of docking locations contained in each set. Out of 15 docking locations, four are represented in the test set, while 14 are represented in the training set. Thus, as every docking location has its own set of challenges with respect to geographical layout and environmental conditions, the ship operator needs to adapt his or her docking strategy. This results in a larger spread in terms of thruster commands, which in turn affects the prediction accuracy of the vessel model.

V. Conclusion

Predicting the motion of a ship is complex. As a way to reduce the uncertainty of the position predictor performance, predictions originating from the vessel dynamic model were combined with a data-based predictor. The latter was implemented using the LSTM neural network methodology. This resulted in a hybrid predictor, where the data-based LSTM corrected the predictions made by the vessel model. A substantial increase in average accuracy was observed throughout the prediction interval. At the maximum prediction horizon of 30 seconds, the average distance error in the position predictions was reduced by about 4 m, from 8.9 m (vessel model) to 4.7 m (hybrid model). Although the black-box nature of the LSTM does not allow for direct insight into what causes the vessel model predictions to deviate, it compensates for the deviations, producing more accurate predictions when both predictors are combined.

TABLE IV
THE AVERAGE POSITION PREDICTION ERROR MADE BY THE VESSEL MODEL PREDICTOR.

t_h [s]	Training error [m]	Testing error [m]
10	3.04	2.34
20	7.11	5.23
30	12.11	8.85

The current study applies prediction solely to provide additional information for the ship operator while docking. Utilizing the proposed hybrid position prediction as input to an automatic motion controller could improve the efficiency and accuracy of autonomous docking operations. Providing a hybrid predictor that meets the two-sided goal of maintaining stability of the cascaded predictor-controller system, as well as to accurately predict the vessel dynamics, would be a key challenge. Along with the inclusion of wind predictions into the hybrid predictor, this constitutes the direction of future work.

Acknowledgement

This work was supported in part by a grant from the Knowledge-Building Project for Industry "Digital Twins for Vessel Life Cycle Service" (Project 280703) and in part by a grant from the Research-Based Innovation "SFI Marine Operation in Virtual Environment" (Project 237929) in Norway. The third author was partially funded by the Norwegian Research Council (NTNU AMOS) at the Norwegian University of Science and Technology (grant no. 223254).

References

[1] X. Yang, "Displacement motion prediction of a landing deck for recovery operations of rotary UAVs," *Int. J. Control. Autom. Syst.*, vol. 11, no. 1, pp. 58–64, 2013.

[2] P. From, J. Gravdahl, and P. Abbeel, "On the influence of ship motion prediction accuracy on motion planning and control of robotic manipulators on seaborne platforms," in *Int. Conf. Robot. Autom.*, 2010, pp. 5281–5288.

[3] S. Küchler, T. Mahl, J. Neupert, K. Schneider, and O. Sawodny, "Active control for an offshore crane using prediction of the vessels motion," *IEEE/ASME Trans. Mechatronics*, vol. 16, no. 2, pp. 297–309, 2011.

[4] T. Praczyk, "Using evolutionary neural networks to predict spatial orientation of a ship," *Neurocomputing*, vol. 166, pp. 229–243, 2015.

[5] Y. Shuai, G. Li, X. Cheng, R. Skulstad, J. Xu, H. Liu, and H. Zhang, "An efficient neural-network based approach to automatic ship docking," *Ocean Eng.*, vol. 191, no. Available online, 2019.

[6] L. P. Perera, "Navigation vector based ship maneuvering prediction," *Ocean Eng.*, vol. 138, pp. 151–160, 2017.

[7] M. Triantafyllou, M. Bodson, and M. Athans, "Real time estimation of ship motions using Kalman filtering techniques," *IEEE J. Ocean. Eng.*, vol. OE-8, no. 1, pp. 9–20, 1983.

[8] S. Sutulo, L. Moreira, and C. Guedes Soares, "Mathematical models for ship path prediction in manoeuvring simulation systems," *Ocean Eng.*, vol. 29, no. 1, pp. 1–19, 2001.

[9] M. M. Sidar and B. F. Doolin, "On the Feasibility of Real-Time Prediction of Aircraft Carrier Motion at Sea," *IEEE Trans. Automat. Contr.*, vol. 28, no. 3, pp. 350–356, 1983.

[10] U. D. Nielsen, A. H. Brodtkorb, and J. J. Jensen, "Response predictions using the observed autocorrelation function," *Mar. Struct.*, vol. 58, no. September 2017, pp. 31–52, 2018.

[11] L. P. Perera and C. Guedes Soares, "Ocean Vessel Trajectory Estimation and Prediction Based on Extended Kalman Filter," in *Adapt. 2010, Second Int. Conf. Adapt. Self-Adaptive Syst. Appl.*, 2010, pp. 14–20.

[12] L. P. Perera, P. Oliveira, and C. Guedes Soares, "Maritime Traffic Monitoring Based on Vessel Detection, Tracking, State Estimation, and Trajectory Prediction," *IEEE Trans. Intell. Transp. Syst.*, vol. 13, no. 3, pp. 1188–1200, 2012.

[13] Y. M. Yin, H. Y. Cui, M. Hong, and D. Y. Zhao, "Prediction of the vertical vibration of ship hull based on grey relational analysis and SVM method," *J. Mar. Sci. Technol.*, vol. 20, no. 3, pp. 467–474, 2014.

[14] M. W. Li, J. Geng, D. F. Han, and T. J. Zheng, "Ship motion prediction using dynamic seasonal RvSVR with phase space reconstruction and the chaos adaptive efficient FOA," *Neurocomputing*, vol. 174, pp. 661–680, 2016.

[15] W. Zhang and Z. Liu, "Real-time ship motion prediction based on time delay wavelet neural network," *J. Appl. Math.*, vol. 2014, 2014.

[16] Z. Peng, J. Wang, and D. Wang, "Distributed Containment Maneuvering of Multiple Marine Vessels via Neurodynamics-Based Output Feedback," *IEEE Trans. Ind. Electron.*, vol. 64, no. 5, pp. 3831–3839, 2017.

[17] ——, "Distributed Maneuvering of Autonomous Surface Vehicles Based on Neurodynamic Optimization and Fuzzy Approximation," *IEEE Trans. Control Syst. Technol.*, vol. 26, no. 3, pp. 1083–1090, 2018.

[18] J. Yin, W. Zhang, T. Li, and J. Hu, "Modified minimal resource allocating network for ship motion predictive control," in *Proc. 2010 Int. Conf. Intell. Control Inf. Process. ICICIP 2010*, no. PART 1, 2010, pp. 231–235.

[19] J. C. Yin, Z. J. Zou, and F. Xu, "On-line prediction of ship roll motion during maneuvering using sequential learning RBF neuralnetworks," *Ocean Eng.*, vol. 61, pp. 139–147, 2013.

[20] 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 Robot. Autom. Mag.*, vol. 26, no. 3, pp. 39–51, 2019.

[21] Q. Xu, X. Li, and C.-y. Chan, "Enhancing Localization Accuracy of MEMS-INS/GPS/In-Vehicle Sensors Integration During GPS Outages," *IEEE Trans. Instrum. Meas.*, vol. 67, no. 8, pp. 1–13, 2018.

[22] L. Chen and J. Fang, "A hybrid prediction method for bridging GPS outages in high-precision POS application," *IEEE Trans. Instrum. Meas.*, vol. 63, no. 6, pp. 1656–1665, 2014.

[23] J. Ma, T. Li, and G. Li, "Comparison of Representative Method for Time Series Prediction," in *2006 Int. Conf. Mechatronics Autom.*, 2006, pp. 2448–2453.

[24] Z. Xiao, L. Ponnambalam, X. Fu, and W. Zhang, "Maritime Traffic Probabilistic Forecasting Based on Vessels' Waterway Patterns and Motion Behaviors," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 11, pp. 3122–3134, 2017.

[25] S. Gan, S. Liang, K. Li, J. Deng, and T. Cheng, "Long-Term Ship Speed Prediction for Intelligent Traffic Signaling," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 1, pp. 82–91, 2017.

[26] J. M. Giron-Sierra and S. Esteban, "The problem of quiescent period prediction for ships: A review," in *IFAC Proc. Vol.*, vol. 43, no. 20. IFAC, 2010, pp. 307–312.

[27] J. G. Kusters, K. L. Cockrell, B. S. Connell, J. P. Rudzinsky, and V. J. Vinciullo, "FutureWaves^{TM}: A real-time Ship Motion Forecasting system employing advanced wave-sensing radar," in *Ocean. 2016 MTS/IEEE Monterey, OCE 2016*, 2016.

[28] E. Soares, P. Costa, B. Costa, and D. Leite, "Ensemble of evolving data clouds and fuzzy models for weather time series prediction," *Appl. Soft Comput. J.*, vol. 64, pp. 445–453, 2018.

[29] N. Sapankevych and R. Sankar, "Time series prediction using support vector machines: A survey," *IEEE Comput. Intell. Mag.*, vol. 4, no. 2, pp. 24–38, 2009.

[30] S. Lefèvre, D. Vasquez, and C. Laugier, "A survey on motion prediction and risk assessment for intelligent vehicles," *ROBOMECH J.*, vol. 1, no. 1, pp. 1–14, 2014.

[31] T. I. Fossen, *Handbook of Marine Craft Hydrodynamics and Motion Control*. John Wiley and Sons Ltd., 2011.

[32] O. N. Smogeli, "Control of Marine Propellers: From Normal to Extreme Conditions," Ph.D. dissertation, Norwegian University of Science and Technology, 2006.

[33] J. Schmidhuber, "Deep Learning in neural networks: An overview," *Neural Networks*, vol. 61, pp. 85–117, 2015.

[34] J. Snoek, H. Larochelle, and R. Adams, "Practical Bayesian Optimization of Machine Learning Algorithms." *Adv. Neural Inf. Process. Syst.*, pp. 2951–2959, 2012.

[35] S. G. Soares and R. Araújo, "An adaptive ensemble of on-line Extreme Learning Machines with variable forgetting factor for dynamic system prediction," *Neurocomputing*, vol. 171, pp. 693–707, 2016.

D

Paper IV

Modeling, Identification and Control, Vol. 42, No. 1, 2021, pp. 17–26, ISSN 1890–1328

A Co-operative Hybrid Model For Ship Motion Prediction

R. Skulstad [1] G. Li [1] T. I. Fossen [2] T. Wang [1] H. Zhang [1]

[1] *Department of Ocean Operations and Civil Engineering, Intelligent Systems Lab, Norwegian University of Science and Technology, N-6002 Aalesund, Norway. E-mail:* {robert.skulstad,guoyuan.li,tongtong.wang,hozh}@ntnu.no

[2] *Department of Engineering Cybernetics , Norwegian University of Science and Technology, N-7491 Trondheim, Norway. E-mail:* thor.fossen@ntnu.no

Abstract

Dynamic models of ships have been widely used for model-based control and short-term prediction in the past. Identifying the parameters of such models has mainly been done through scaled model tests, full scale tests or computational fluid dynamics software. This is a challenging task due to the many aspects that influence the ship dynamic behaviour and thus one would expect a certain degree of mismatch between the actual motion of the ship and the modelled behaviour. The mismatch in the dynamic model may be due to unmodelled effects, but also the lack of measurements of waves and ocean current. To make up for the discrepancies the authors propose to create a co-operative hybrid model consisting of the dynamic model and a neural network, where the neural network predicts the acceleration error of the dynamic model. The approach is tested on real data originating from the Research Vessel (RV) Gunnerus performing a shutdown of thrusters during stationkeeping. The subsequent task is to predict the propagation of position and heading while drifting due to wind, wave and current forces. Comparing the motion of the real ship and the modelled ship, shows the improved prediction accuracy of the hybrid model.

Keywords: Ship motion prediction, Hybrid model, Dynamics

1 Introduction

Short-term ship motion prediction, ranging in prediction interval length from a few seconds to one minute, has implications for a variety of applications such as takeoff and landing of unmanned aerial vehicles and helicopters Yang (2013), ship crane operations From et al. (2010) Küchler et al. (2011), maneuvering Perera (2017) and quiescent period prediction Dannenberg et al. (2010) Giron-Sierra and Esteban (2010). The main motivating factors for all these applications of ship motion prediction are safety and efficiency. If the uncertainty of the future motion is reduced, the ship operator can make informed decisions on how to proceed with an operation. Helicopter landing may be timed to coincide with a period of relatively low heave motions or a trajectory that may interfere with surrounding vessels or stationary objects may be discovered at an early stage and corrective measures can be implemented.

The motion of a ship at sea is dictated by the hydrodynamic properties of the ship, environmental disturbances like wind, wave and ocean current and thruster forces. Typically, the wind is the only measurable environmental state, which leaves the wave impact and current impact on the model unmeasured. Probably, this is why many researchers have turned to data-based modelling for predicting ship motion. It allows for fitting models based on informative features from recently sampled ship motion data. If the ship is performing Dynamic Positioning (DP), meaning that it maintains a steady position, or performs low-speed ma-

doi:10.4173/mic.2021.1.2

neuvering using thrusters, observers have been developed that filter out the wave-induced oscillatory motion and estimate the apparent force caused by a combination of wave drift force and ocean currents Fossen and Perez (2009). However, they depend on information about the sea state and requires that the ship is actively controlled.

In this paper we investigate the benefit of combining a dynamic model with a data-based model for predicting the future position and heading of a ship. The dynamic model is identified through scaled model experiments and describes the motion of the ship in 3 degrees of freedom (DOF): longitudinal/lateral motion and rotation about the vertical axis of the ship Hassani et al. (2015). Forces due to wind and thrusters are accounted for as well. But, due to lack of knowledge of wave and current impact on the ship and inherent model fidelity limitations, the model output will not describe the 3 DOF motion of the real ship perfectly. Thus, based on a limited set of recent data to perform model parameter adaptation, the data-based model corrects the dynamic model on an acceleration level. In this way, the general trends in future motion may be outlined by using existing knowledge, leaving the data-based model with less of a challenge in predicting the residuals of the original model.

Hybrid modelling, using an identified model of a process along with a data-based model to amend its deficiencies, has been explored in the past van de Ven et al. (2007). Albeit, with a focus of improving parameter estimates of a partially known model. The present study focuses on compensating the predictions made by a model which is assumed to be complete, but lacking information about the complete environmental disturbances Skulstad et al. (2021).

When the ship is automatically controlled, either for stationkeeping or trajectory tracking under normal operating conditions, the assumption is that the ship will maintain its desired motion state and therefore no prediction is needed. Therefore, the case study presented in this paper shows the use of predictions during a power failure situation. While power failures during stationkeeping operations is unlikely due to power plant and thruster redundancy for ships involved in critical operations, the consequences may be severe. Often such operations take place close to other ships or offshore structures or during deployment of seabed installations. Being able to predict the future position and heading during such a failure could provide the ship operator with valuable information on whether or not to carry out the operation, and thus enhances the quality of the pre-operation risk analysis. This insight might also be of use in decision-making for autonomous ships where a failure requires a proper response based on the future trajectory of the ship Blindheim et al. (2020).

2 Related work

Ship motion prediction models range from completely transparent kinetic models Triantafyllou et al. (1983), through kinematic models Perera (2017) to regression models Brandsæter and Vanem (2018) and black-box Machine Learning (ML) models Yin et al. (2017). Each domain has their own strengths and weaknesses. Applying a kinetic model to prediction requires knowing the parameters that go into the model. This may be performed using scaled model experiments Hassani et al. (2015) or specialized hydrodynamic computer programs. It is challenging to determine all the parameters of the models to a satisfactory accuracy van de Ven et al. (2007) covering all the various speed regimes and environmental conditions a vessel may encounter. However, the relations between measured data and future motion are explicit and defined by functions describing forces that are derived using well-established theory Fossen (2011).

Kinematic models disregard the forces induced by a ship moving on the surface of the ocean and apply only the relation between acceleration/velocity and heading to get positions. A complete method of estimation and prediction of vessel trajectories is presented in Perera (2017). An extended Kalman filter was used to estimate the states of a kinematic ship maneuvering model. Estimated states were then applied to determine navigation vectors, which were input to a vector product-based prediction method for calculating future positions and heading.

ML offers a way of modelling the ship behaviour without explicitly identifying parameters that relate environmental disturbances, ship state and thruster forces to future ship states Li et al. (2017). However, this comes at the cost of model transparency and the requirement of having sufficiently rich data such that the data-based model is able to generalize to new inputs. An online approach to ship roll angle prediction was presented in Yin et al. (2017). They applied an adaptive sliding window to include relevant recently sampled data which best describe the time-varying dynamics of the ship. The Radial Basis Function (RBF) Neural Network (NN) used in their study was updated in a sequential manner. A similar sequential RBF network was used for multi-step predictions in relation to predictive control of a ship's course in Yin et al. (2010). Recently, recurrent networks have also been used for predicting roll/pitch angles and heave motion Zhang et al. (2020) Duan et al. (2019) and horizontal motion Skulstad et al. (2019).

Support vector regression (SVR) models represent an alternative to NNs for creating data-based predictive models Li et al. (2016). By recognizing that the roll angle exhibits a periodic term when exposed to waves, Li et al. combined a periodogram estimation method with a SVR model to predict the roll angle up to 15 seconds ahead Li et al. (2019). Comparison against pure SVR and NN models showed the benefit of including such a hybrid prediction scheme. The periodicity seen in the time series of ship roll and pitch angle and heave motion may also be handled by decomposition of the signals using e.g. empirical mode decomposition Hong et al. (2019) Duan et al. (2015b). For such an approach each mode is related to a predictive model, such as the SVR, and the individual predictions are merged to form the prediction of the original signal.

The Autoregressive (AR) method makes use of only the history samples of a certain state and is an efficient way of obtaining predictions a few seconds ahead From et al. (2010). Optimized parameters may be obtained through a recursive least squares method Ma et al. (2006). The AR method is efficient, but it is a linear method which is limited by the assumption of having a stationary time series Duan et al. (2015a). To overcome these limitations Yang et al. applied the Bayes Information Criteria to select the model coefficient size and also implemented a factor that reduced the impact of the most distant samples Yang (2013).

The performance of data-based methods, such as ML and AR, heavily depend on the amount and relevance of the training data with respect to the data used during the prediction stage. And, given a certain training dataset containing relevant and sufficient amount of data, the training time can not be determined up front Takami et al. (2021). Predictors that apply a priori information about the ship dynamics, such as identified maneuvering models, face fidelity issues due to the abundance of operational conditions that influence on the dynamics of the ship. The deficiencies of the two modelling domains makes combining models from the two a natural choice. Training efficiency of the data-based predictor may be enhanced by predicting residual errors in the dynamic model predictions. And those residual predictions may be used to account for the previously mentioned fidelity issues Skulstad et al. (2021).

3 Ship motion predictors

In this section predictors that apply a priori information about the components of the system will be introduced. In a scenario where data is scarce and the event that triggers the prediction is seldom seen, purely data-based predictors will have limited information to perform successful predictions.

3.1 Model-based predictor

A purely model-based ship motion predictor embeds the models of the environment impact on the ship, thruster forces and the hydrodynamic models of the ship hull. This is visualized in Figure 1.

In combination these models output the total force exerted on the ship hull and these forces are converted to accelerations through the known mass matrix of the ship, shown in eq. (1).

$$\boldsymbol{M}_{RB}\dot{\boldsymbol{\nu}} + \boldsymbol{C}_{RB}(\boldsymbol{\nu})\boldsymbol{\nu} + \boldsymbol{M}_A\dot{\boldsymbol{\nu}}_r + \boldsymbol{C}_A(\boldsymbol{\nu}_r)\boldsymbol{\nu}_r + \boldsymbol{D}\boldsymbol{\nu}_r + \boldsymbol{D}_n(\boldsymbol{\nu}_r)\boldsymbol{\nu}_r = \boldsymbol{\tau}_c + \boldsymbol{\tau}_{wi} + \boldsymbol{\tau}_{wa}. \quad (1)$$

In the above equation the ship velocity relative to current is given as $\boldsymbol{\nu}_r = \boldsymbol{\nu} - \boldsymbol{\nu}_c$ where $\boldsymbol{\nu}_c = \begin{bmatrix} u_c & v_c & 0 \end{bmatrix}^T$ denotes the components of the current velocity in the coordinate frame of the ship. The current velocity is presumed irrotational in this case. $\boldsymbol{M}_{RB}$ and $\boldsymbol{M}_A$ are the rigid body and added mass matrices of the ship. Coriolis and centripetal forces are included through $\boldsymbol{C}_A(\boldsymbol{\nu}_r)$ and $\boldsymbol{C}_{RB}(\boldsymbol{\nu})$, linear damping through $\boldsymbol{D}$ and nonlinear damping through $\boldsymbol{D}_n(\boldsymbol{\nu}_r)$. Forces generated by the three thrusters are given as $\boldsymbol{\tau}_c$, while the environmental forces due to wind and waves are given by $\boldsymbol{\tau}_{wi}$ and $\boldsymbol{\tau}_{wa}$, respectively. $\boldsymbol{\tau}_{wa}$ and $\boldsymbol{\nu}_c$ are unknown and will therefore not be considered further as inputs to the dynamic model.

The block named *Model-based predictor* in Figure 1 contains these models. Integrating the accelerations once yields the predicted propagated velocities, $\hat{\boldsymbol{\nu}}_{t+1}$, given in the coordinate frame of the ship. A rotation by the heading angle to obtain velocities relative to North and East, followed by a second integration results in predicted positions and heading, $\hat{\boldsymbol{\eta}}_{t+1}$.

At the beginning of the prediction interval the thrusters are each fed with their respective most recent sample from the thrusters, u_0, where the subscript 0 signifies the start of the prediction interval. For the azimuth thrusters this corresponds to an azimuth angle and a Revolutions Per Minute (RPM) command, and for the tunnel thruster an RPM command. During a power failure the thrusters do not actively produce force. The only force they induce on the ship is due to moving through the water at a non-zero velocity. These values are updated based on the predicted ship velocities.

Thruster forces are acquired through the execution of hydrodynamic models created by the manufacturer of the thrusters. The individual thrust outputs depend on the control commands issued by the motion

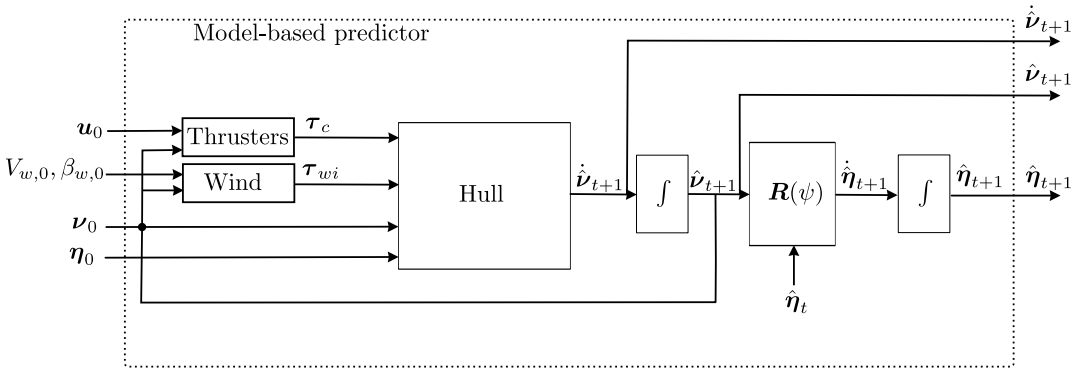


Figure 1: Model-based predictor

controller as well as the ship speed and is valid for the 4 quadrants of operation given in Table 1 Smogeli (2006).

Table 1: The 4 quadrants of propeller operation parameterized by RPM (n) and inflow velocity (V_a) (courtesy of Smogeli (2006)).

Parameter	$\mathbf{1}^{st}$	$\mathbf{2}^{nd}$	$\mathbf{3}^{rd}$	$\mathbf{4}^{th}$
n	≥ 0	< 0	< 0	≥ 0
V_a	≥ 0	≥ 0	< 0	< 0

The ship used in this study has three thrusters: one bow tunnel thruster and two main azimuth thrusters at the stern. Converting their individual, local, thrust force into a combined force acting in the centre of gravity of the ship requires the 3x3 thrust configuration matrix shown in eq. (2). $s(\cdot)$ and $c(\cdot)$ are abbreviations of the sine and cosine functions.

$$\boldsymbol{\tau}_c = \begin{bmatrix} 0 & c(\alpha_p) & c(\alpha_s) \\ 1 & s(\alpha_p) & s(\alpha_s) \\ l_{tx} & l_{px}s(\alpha_p) & l_{sx}s(\alpha_s) \\ & -l_{py}c(\alpha_p) & -l_{sy}c(\alpha_s) \end{bmatrix} \times \begin{bmatrix} T_{tn} \\ T_{pa} \\ T_{sa} \end{bmatrix} \tag{2}$$

The tunnel thruster cannot rotate and thus produces positive or negative thrust, T_{tn}, along the lateral axis of the ship at a distance l_{tx} from the ship's centre of gravity. Azimuth thrusters may rotate and the angles for the port and starboard thruster are α_p and α_s, respectively. Their individual thrust, T_{pa} and T_{sa}, is applied at locations (l_{px},l_{py}) and (l_{sx},l_{sy}) relative to the ship's centre of gravity.

Wind related variables are also unknown for future time instances, so the values sampled at the prediction interval start is used ($V_{w,0}$ and $\beta_{w,0}$). The resulting estimated force, $\boldsymbol{\tau}_{wi}$, given in the coordinate frame of the ship is shown in eq. (3).

$$\boldsymbol{\tau}_{wi} = \frac{1}{2}\rho_a V_{rw}^2 \begin{bmatrix} C_X(\gamma_{rw})A_{Fw} \\ C_Y(\gamma_{rw})A_{Lw} \\ C_N(\gamma_{rw})A_{Lw}L_{oa} \end{bmatrix} \tag{3}$$

The wind force coefficients C_X, C_Y and C_N depend on the relative wind direction, γ_{rw}, and they are typically derived using either computational fluid dynamics software or wind tunnel tests Hassani et al. (2015). In addition to the force coefficients the model requires air density, ρ_a, relative wind velocity, V_{rw}, area of the ship's frontal and lateral projection, A_{Fw} and A_{Lw}, and the overall length of the ship, L_{oa}.

To perform multi-step predictions the equation of motion, given in eq. (1), is solved for the acceleration, $\dot{\boldsymbol{\nu}}$. Integration according to Figure 1 allows for feeding back the velocities. Repeating this process gives predicted accelerations, velocities, positions and heading at future time instances.

3.2 Hybrid model predictor

In the hybrid model the dynamic model is applied as described in section 3.1, while a data-based model aims to predict the discrepancy in the acceleration of the model-based predictor. Discrepancies will always be present due to the complex environment in which the ship operates. In the present study the discrepancies also come from not having measurements or estimates of the ocean current, such that the dynamic model may account for these forces. The training scheme is shown in Figure 2.

The objective of the NN predictor is to predict $\Delta\dot{\boldsymbol{\nu}}_{t+1} = \dot{\boldsymbol{\nu}}_{t+1} - \dot{\hat{\boldsymbol{\nu}}}_{t+1}$, where $\dot{\boldsymbol{\nu}}_{t+1}$ is the actual acceleration of the ship and $\dot{\hat{\boldsymbol{\nu}}}_{t+1}$ is the acceleration predicted by the model-based predictor. For each sample in the *Historical data* of Figure 2, the model-based

prediction of the surge, sway and yaw acceleration one second ahead is compared against the true acceleration one second ahead. This yields the targets for the supervised training procedure of an ensemble of feedforward NNs. Ensembles are applied in order to average the effect of random weight/bias initialization in the individual NNs. A subset of all the available inputs, shown as $\boldsymbol{X}_{tr,t}$ in Figure 2, is extracted according to the feature selection procedure described in Section 3.3. The available features also include virtual sensors in the form of states predicted by the model-based predictor and the NNs (see Figure 2). This results in the input training data, $\boldsymbol{X}'_{tr,t}$.

Figure 2 shows the application of the hybrid predictor when used to output predictions of the future north/east position and heading angle. The model-based predictor initially receives the following, most recently sampled, data:

- Position and heading, $\boldsymbol{\eta}_0$, relative to the North East Down coordinate frame.
- Linear and angular velocities, $\boldsymbol{\nu}_0$, given in the coordinate frame of the ship.
- Wind speed and direction, $V_{wi,0}$ and $\beta_{wi,0}$ (relative to north).
- Thruster control commands, $\boldsymbol{u}_t$. These remain constant throughout the prediction interval.

Executing the model-based predictor results in predictions of acceleration, velocity and position at one second ahead of the prediction interval start time. These predictions are passed to the *Sensors* block for potential use as input features to the NNs at subsequent steps. Then an input vector is drawn from the set of available sensor data according to the feature selection scheme. A predicted acceleration discrepancy is output by a forward pass over the data-based predictor to get $\Delta\hat{\dot{\nu}}_{t+1}$. Summing the predicted acceleration discrepancy and the model-based predicted accelerations yields the corrected acceleration, which is propagated through integrators to achieve the predicted position and heading angle at one second ahead of the initial time instance. This process is repeated 60 times for a one minute prediction interval.

A motivating factor for selecting an iterative multi-step prediction strategy, as opposed to directly predicting the complete future acceleration discrepancy vector, lies in the nature of the process that generates the training data. When the ship is actively controlled the control commands change as fast as 1 Hz. This limits the temporal validity of the acceleration predictions made by the model-based predictor, which is used to generate supervised training data for the NNs. The NN structure applied in this paper is a regular feedforward network consisting of two layers of 20 units applying a hyperbolic tangent activation function. The output layer has three units and applies linear activation functions.

3.3 Feature selection and extraction

Selecting features that contain useful information for predicting a certain target value is beneficial in terms of generalization ability of the NNs, reducing computational burden and providing a more interpretable model. In this paper this is achieved by first acknowledging that as the model-based predictor performs multi-step predictions, only the position and heading and the respective velocities will be propagated/updated. Thus, features such as roll and pitch angle, their velocities, and wind direction and velocity remain constant throughout the prediction interval. Also, the control commands to thrusters remain constant as they do no longer contribute to actively control the ship's position and heading. By this intuition, only the position, heading and their velocities and accelerations will provide current information about the motion of the ship within the prediction interval. From this subset, which contains features that are dynamic in the prediction interval, the model-predicted longitudinal, lateral and rotational speed and accelerations were selected as input to the data-based predictor along with the previously predicted acceleration discrepancy, $\Delta\hat{\dot{\boldsymbol{\nu}}}_t$. The NNs thereby assume an auto-regressive form with exogenous inputs.

4 Case study

In this section the performance of the predictors are compared in a scenario where the ship operator turns off the active dynamic positioning controller. The scenario was conducted outside the port of Trondheim, Norway, and the purpose was to mimic a power failure event. The analysis, training and application of the predictors is carried out based on historical data, i.e. not onboard the ship in real time.

4.1 Data and ship model

The ship used in this case study is the Research Vessel (RV) Gunnerus shown in Figure 3. It is equipped with two azimuth thrusters at the stern and one bow tunnel thruster and has an overall length of 36.25 m.

The data used for the case study presented in this section was collected during a shutdown of the active dynamic positioning control, leaving the vessel to drift freely due to forces incurred by current, wind and

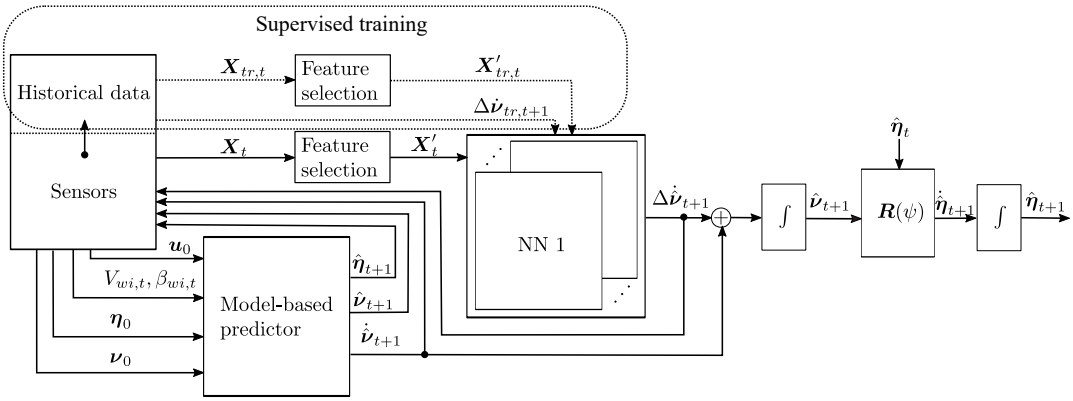


Figure 2: Hybrid predictor

waves. The sea state during the data collection was estimated by the ship operator to correspond to a sea state of 2 on the Beaufort scale. Prior to executing the simulated shutdown of dynamic positioning control, the ship performed low-speed maneuvering and stationkeeping for 50 minutes, sampling data at 1 Hz from thrusters, ship motion data, wind direction and wind speed.

Figure 3: Starboard view of the RV Gunnerus.

4.2 Results

Since the parameters of the ship- and thruster models have been derived through tests using scaled model experiments and computational fluid dynamics software they describe the dynamic behaviour of the overall system well. Applying the models as-is therefore yields a reasonably accurate set of predicted future position and heading. Figure 4 shows the velocities predicted every second from the point of failure to the prediction horizon end at 60 seconds ahead of time.

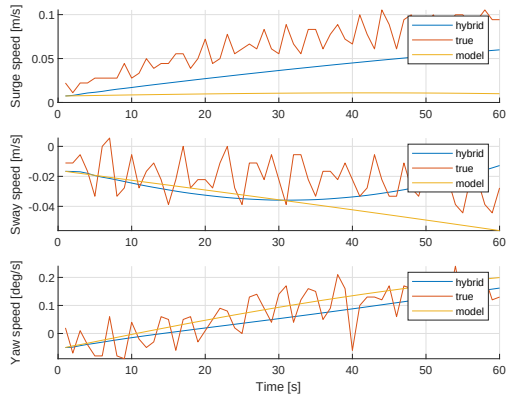


Figure 4: The velocities related to the first prediction interval starting from the point of failure.

The performance of the model-based predictor is sensitive to initial velocities and heading. To mitigate the effect of noisy velocity measurements, seen as red lines in Figure 4, on the model-based predictions, a sliding window average of 3 seconds was applied to the heading and velocities input at the start of each prediction interval. A larger discrepancy is seen for the surge velocity predicted by the dynamic model, which indicates that either wave/ocean current-induced forces, or model discrepancies are present. The smooth output from the model-based predictor, in addition to reliably propagating its predictions, has the added effect of smoothing the output of the hybrid predictor (see Figure 5).

Figure 5 shows the acceleration prediction made by the two predictors over the course of the first predic-

Table 2: Average errors in terms of North/East position and yaw angle for the first 5 and 20 prediction intervals after the power failure.

Model type	State	Position and Heading error (5/20 intervals)			
		15 s	30 s	45 s	60 s
Hybrid model	North	0.14/0.17	0.22/0.76	0.41/1.74	0.70/2.93
	East	0.06/0.07	0.16/0.36	0.38/1.07	0.99/2.41
	Yaw	0.14/0.21	0.28/0.56	0.37/0.99	0.73/1.71
Dynamic model	North	0.26/0.21	0.73/0.56	1.36/1.07	2.12/1.78
	East	0.11/0.12	0.47/0.48	1.05/1.01	1.79/1.67
	Yaw	0.25/0.23	0.55/0.56	0.89/0.71	1.49/0.95

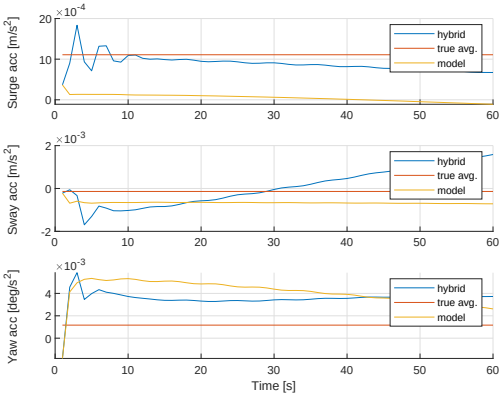


Figure 5: The accelerations related to the first prediction interval starting from the point of failure. Note that the actual acceleration (red line) is given as the average of the 60 second prediction interval.

tion interval, starting from the time of failure. The closer proximity of the hybrid predicted acceleration to the mean acceleration, given in red, shows the positive effect of adding the acceleration prediction compensation. By running multiple prediction intervals, successively incrementing the start of the interval with one second, the average prediction errors may be discerned. In Table 2, average values of the first 5 and 20 successive, post failure, prediction intervals are given.

Due to the nature of ML algorithms, requiring data for inference which is similar to that seen in training, the predictions made on data sampled some time after the failure may not be valid. This is caused by the ship drifting and thus obtaining higher velocities relative to the low-speed and stationkeeping operation from which the training data was sampled. This is reflected by the degradation in performance of the hybrid predictor relative to the model-based predictor as the number of prediction intervals increases (see Table 3). If a set of data covering a larger dynamic range was pre-recorded, the validity of the hybrid predictions may be extended beyond what is shown in this paper. Unlike the hybrid predictor, the average error of the model-based predictor decreases as the prediction interval start time is propagated (see Table 2). This is attributed to the fact that as the ship drifts the transient accelerations, caused by the sudden loss of stationkeeping control, decrease.

Table 3: Average reduction in error by applying the hybrid model relative to the dynamic model predictor.

State	Error reduction [%] (5/20 intervals)			
	15 s	30 s	45 s	60 s
North	46/14	66/-35	70/-62	67/-65
East	45/67	66/25	64/-6	45/-44
Yaw	44/9	49/0	58/-39	51/-80

When the ship no longer has active stationkeeping capabilities, it drifts northwards for this specific case. This is shown in Figure 6 where the black dashed lines of the figure show the trajectory of the first 5 prediction intervals of the hybrid predictor, while the dash-dot lines show the corresponding model-based predictions. Averages of the 5 trajectories are given as red circles (hybrid model) and red crosses (dynamic model). The initial speed direction for the first prediction interval is given by the black arrow. Variations in the predicted trajectories are mainly due to the advancing prediction interval start time, which yields varying initial speeds. The average of the trajectories shown in Figure 6 clearly show the benefit of including the acceleration corrections.

The measured wind direction and speed for the cor-

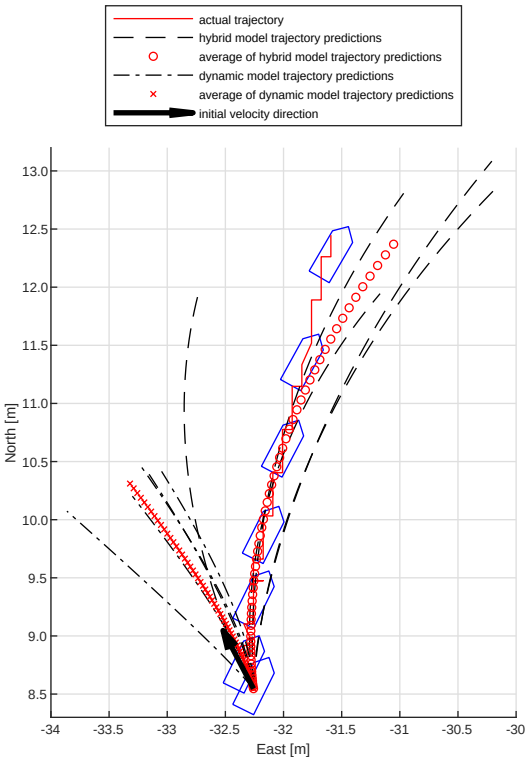


Figure 6: Trajectories of the ship while drifting. The blue ship frames, plotted every 10 seconds, indicate the actual heading angle of the ship.

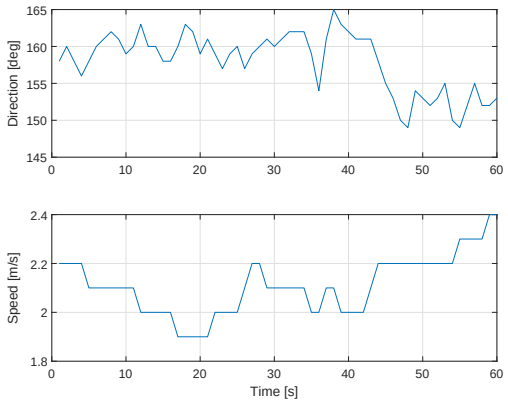


Figure 7: Actual wind speed and direction relative to north (0 degrees). The direction is to be interpreted as "coming from ".

responding time interval are shown in Figure 7. The mean direction is roughly 157 degrees. Thus, wind effect induces forces on the hull consistent with the drifting northward trajectory. However, the ship also moves slightly due east, which is not consistent with the wind direction. Thus, due to the modest wave height, current is the remaining factor that may lead to this trajectory.

5 Conclusion

A hybrid predictor was constructed and its predictive performance was tested on data from a real scenario mimicking a power failure during dynamic positioning. The NN models providing acceleration corrections to the model were shown to substantially improve the accuracy of the overall motion predictions for such a case. However, as the ship drifts freely due to wind and current, it eventually attains speeds outside of the range of the data that was used to train the NNs. This limits the validity of such corrective predictions to a short time span after experiencing such a failure. In the case study presented in this paper a good match between the predicted position and heading compared to the actual position and heading was seen for the first 10, 60-second, prediction intervals.

The proposed approach may be transferred to more general purpose prediction scenarios. However, the accuracy of this method, for scenarios where future thruster commands may not be deduced at the time of prediction start, will be reduced. This is due to the influence of commands, not yet known to the predictor.

Acknowledgments

This work was supported in part by a grant from the Knowledge-Building Project for Industry "Digital Twins for Vessel Life Cycle Service" (Project 280703) and in part by a grant from the Research-Based Innovation "SFI Marine Operation in Virtual Environment" (Project 237929) in Norway. The third author was partially funded by the Norwegian Research Council (NTNU AMOS) at the Norwegian University of Science and Technology (grant no. 223254).

References

Blindheim, S., Gros, S., and Arne, T. Risk-Based Model Predictive Control for Autonomous Ship Emergency Management. *IFAC-PapersOnLine*, 2020. doi:10.1016/j.ifacol.2020.12.1456.

Brandsæter, A. and Vanem, E. Ship speed prediction based on full scale sensor measurements of shaft thrust and environmental conditions. *Ocean Engineering*, 2018. 162(May):316–330. doi:10.1016/j.oceaneng.2018.05.029.

Dannenberg, J., Hessner, K., Naaijen, P., van den Boom, H., and Reichert, K. The On board Wave and Motion Estimator OWME. In *International Offshore and Polar Engineering Conference*, volume 7. pages 424–431, 2010.

Duan, S., Ma, Q., Huang, L., and Ma, X. A LSTM deep learning model for deterministic ship motions estimation using wave-excitation inputs. *Proceedings of the International Offshore and Polar Engineering Conference*, 2019. 1:959–965.

Duan, W.-y., Huang, L.-m., Han, Y., and Wang, R. IRF - AR Model for Short-Term Prediction of Ship Motion. In *Proceedings of the Twenty-fifth (2015) International Ocean and Polar Engineering Conference*. pages 59–66, 2015a.

Duan, W.-y., Huang, L.-m., Han, Y., Zhang, Y.-h., and Huang, S. A hybrid AR-EMD-SVR model for the short-term prediction of nonlinear and non-stationary ship motion. *Journal of Zhejiang University Science A*, 2015b. 16(7):562–576. doi:10.1631/jzus.A1500040.

Fossen, T. I. *Handbook of Marine Craft Hydrodynamics and Motion Control*. John Wiley and Sons Ltd., 2011. doi:10.1002/9781119994138.

Fossen, T. I. and Perez, T. Kalman Filtering for Positioning and Heading Control of Ships and Offshore Rigs: Estimating the effects of waves, wind, and current. *IEEE Control Systems Magazine*, 2009. 29(6):32–46. doi:10.1109/MCS.2009.934408.

From, P., Gravdahl, J., and Abbeel, P. On the influence of ship motion prediction accuracy on motion planning and control of robotic manipulators on seaborne platforms. In *International Conference on Robotics and Automation*. pages 5281–5288, 2010. doi:10.1109/ROBOT.2010.5509813.

Giron-Sierra, J. M. and Esteban, S. The problem of quiescent period prediction for ships: A review. In *IFAC Proceedings Volumes*, volume 43. pages 307–312, 2010. doi:10.3182/20100915-3-DE-3008.00007.

Hassani, V., Ross, A., Selvik, Ø., Fathi, D., Sprenger, F., and Berg, T. E. Time Domain Simulation Model For Research Vessel Gunnerus. In *International Conference on Ocean, Offshore and Arctic Engineering*. pages 1–6, 2015.

Hong, W. C., Li, M. W., Geng, J., and Zhang, Y. Novel chaotic bat algorithm for forecasting complex motion of floating platforms. *Applied Mathematical Modelling*, 2019. 72:425–443. doi:10.1016/j.apm.2019.03.031.

Küchler, S., Mahl, T., Neupert, J., Schneider, K., and Sawodny, O. Active control for an offshore crane using prediction of the vessels motion. *IEEE/ASME Transactions on Mechatronics*, 2011. 16(2):297–309. doi:10.1109/TMECH.2010.2041933.

Li, G., Kawan, B., Wang, H., and Zhang, H. Neural-network-based modelling and analysis for time series prediction of ship motion. *Ship Technology Research (Schiffstechnik)*, 2017. 64(1):30–39. doi:10.1080/09377255.2017.1309786.

Li, M. W., Geng, J., Han, D. F., and Zheng, T. J. Ship motion prediction using dynamic seasonal RvSVR with phase space reconstruction and the chaos adaptive efficient FOA. *Neurocomputing*, 2016. 174:661–680. doi:10.1016/j.neucom.2015.09.089.

Li, M. W., Geng, J., Hong, W. C., and Zhang, L. D. Periodogram estimation based on LSSVR-CCPSO compensation for forecasting ship motion. *Nonlinear Dynamics*, 2019. 97(4):2579–2594. doi:10.1007/s11071-019-05149-5.

Ma, J., Li, T., and Li, G. Comparison of Representative Method for Time Series Prediction. In *2006 International Conference on Mechatronics and Automation*. pages 2448–2453, 2006. doi:10.1109/ICMA.2006.257735.

Perera, L. P. Navigation vector based ship maneuvering prediction. *Ocean Engineering*, 2017. 138:151–160. doi:10.1016/j.oceaneng.2017.04.017.

Skulstad, R., Li, G., Fossen, T. I., Vik, B., and Zhang, H. Dead Reckoning of Dynamically Positioned Ships: Using an Efficient Recurrent Neural Network. *IEEE Robotics and Automation Magazine*, 2019. 26(3):39–51. doi:10.1109/MRA.2019.2918125.

Skulstad, R., Li, G., Fossen, T. I., Vik, B., and Zhang, H. 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*, 2021. 70. doi:10.1109/TIM.2020.3018568.

Smogeli, O. N. *Control of Marine Propellers: From Normal to Extreme Conditions*. Ph.D. thesis, Norwegian University of Science and Technology, 2006.

Takami, T., Nielsen, U. D., and Jensen, J. J. Real-time deterministic prediction of wave-induced ship responses based on short-time measurements. *Ocean Engineering*, 2021. 221(December 2020). doi:10.1016/j.oceaneng.2020.108503.

Triantafyllou, M., Bodson, M., and Athans, M. Real time estimation of ship motions using Kalman filtering techniques. *IEEE Journal of Oceanic Engineering*, 1983. OE-8(1):9–20. doi:10.1109/JOE.1983.1145542.

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

Yang, X. Displacement motion prediction of a landing deck for recovery operations of rotary UAVs. *International Journal of Control, Automation and Systems*, 2013. 11(1):58–64. doi:10.1007/s12555-011-0157-8.

Yin, J., Wang, N., and Perakis, A. N. A Real-Time Sequential Ship Roll Prediction Scheme Based on Adaptive Sliding Data Window. *IEEE Transactions on Systems, Man, and, Cybernetics: Systems*, 2017. pages 1–11. doi:10.1109/TSMC.2017.2735995.

Yin, J., Zhang, W., Li, T., and Hu, J. Modified minimal resource allocating network for ship motion predictive control. In *Proceedings of 2010 International Conference on Intelligent Control and Information Processing, ICICIP 2010*, PART 1. pages 231–235, 2010. doi:10.1109/ICICIP.2010.5565241.

Zhang, G., Tan, F., and Wu, Y. Ship Motion Attitude Prediction Based on an Adaptive Dynamic Particle Swarm Optimization Algorithm and Bidirectional LSTM Neural Network. *IEEE Access*, 2020. 8:90087–90098. doi:10.1109/ACCESS.2020.2993909.

E

Paper V

Constrained Control Allocation For Dynamic Ship Positioning Using Deep Neural Network

Robert Skulstad, Guoyuan Li, *Senior Member, IEEE*, Thor I. Fossen, *Fellow, IEEE*,
Bjørnar Vik and Houxiang Zhang, *Senior Member, IEEE*

***Abstract*—Dynamic positioning (DP) is one of the key technologies towards ship autonomy. Ships in DP mode use thruster devices to maintain position and perform low-speed maneuvering. The motion controller issues force requests according to the measured ship state and motion objectives. These requests must be translated into individual thruster commands. Due to constraints in the thrusters, such as inertia and limited angles of operation, state-of-the-art control allocation methods apply constrained optimization techniques. Although such methods readily capture the handling of constraints, they may require significant computational resources in searching for optimized commands in real time. Here we show that a neural network may be applied to offer an effective evaluation of the mapping between motion controller requests and executable thruster commands. An Autoencoder-like neural network is trained with data generated using knowledge about the configuration of the thrusters. Custom loss functions shape the weights of the network, such that the overall motion objectives are met and thruster constraints are honored. Then, the network is applied to perform low-speed maneuvering and stationkeeping in a simulator. Comparison relative to a basic fixed-angle allocator indicate similar dynamic performance at a reduced power consumption.**

***Index Terms*—Deep Learning Methods, Motion Control, Constrained Control Allocation**

I. Introduction

CONTROL allocation for normal surface vessels and autonomous ships involves distributing force requests made by a preceding motion controller. The force requests are given relative to the vessel center of gravity and therefore needs to be translated into commands specific to each thrust-producing device, hereafter termed thruster. Typically, ships involved in complex operations have redundant thrusters, which allows for increased safety in case of thruster failures, but also allows for optimizing the overall motion performance in terms of eg. power consumption. Such a thruster configuration renders the ship over-actuated, which implies that more than one set of commands may meet the overall force request. The primary goal of the control allocator is to allocate individual thruster control signals such that the thrusters jointly produce the requested force. Secondary goals include managing the limitations of the thrusters in terms of magnitude and rates, minimization of power consumption, forbidden sectors of operation (rotatable thrusters) and the avoidance of singularities [1].

To solve the constrained control allocation problem for ships, researchers have explored several methods, the most popular one being optimization based methods. Basic approaches to this domain of methods involve posing the problem as an unconstrained optimization problem [2]. This disregards the inherent physical constraints of a thruster, such as force magnitude and its force rate of change. If the thruster can rotate, the maximum/minimum angle and the angle change rate must be considered as well. Thus, this approach may produce infeasible commands, resulting in a failure to meet the force request of the motion controller. Optimization-based approaches have been applied to deal with several challenging problems in control allocation. In terms of maintaining maneuverability when azimuth thrusters that rotate slowly are applied, singularity avoidance is important [3], [4]. This technique avoids thruster configurations where no force can be produced in one of the ship axes. Load variation reduction on marine power plants [5] and fuel minimization of marine power plants [6] have also received focus. This reduces wear and tear on the power production system, emissions and fuel costs. For ships that have fixed main propellers with rudders, the corresponding feasible thrust vector set is non-convex [7]. This is due to the reduced lift force produced by the rudder at negative propeller thrust. Thruster-thruster and thruster-hull hydrodynamic interactions were considered in [8].

In an effort to reduce power consumption and rate changes in the face of environment disturbances, Skjong et al. proposed the use of Model Predictive Control (MPC) to optimize control allocation in the long run [9]. The use of the command horizon, inherent to the MPC domain, allowed for both negating the effect of oscillating wave forces as well as to rotate azimuth thrusters to more favorable directions (ie. thrusters that are more efficient at positive Revolutions Per Minute (RPMs)).

Neural networks used for solving control allocation problems have been researched within the aerospace domain. By posing the control allocation problem as a convex nonlinear program, Chen applied a neural network to solve for the control commands for a near-space vehicle [10]. A more direct approach was taken by Huan et al. to create a control allocator for a space re-entry vehicle [11]. A deep autoencoder-like neural network was applied to model the relationship between the controller requests and the resulting commands to actuators. A similar network structure will be used in this paper. Although the type of layer in the network differs and additional features for constraining the allocated commands are added.

Using neural networks for control allocation directly offers a general purpose tool in terms of not imposing bounds on the convexity or linearity of the proposed system. Neural networks can provide allocation solutions at low latency due to needing only a forward pass over the trained network. However, the training phase may be time consuming, which

dictates offline training. If applied to an autonomous ship, the operating system may invoke the retraining process based on the current operational requirements.

The present paper elaborates on how a Deep Neural Network (DNN) can be constructed for control allocation, along with training design considerations and simulations exploring different aspects of the method. Based on knowledge about the thruster configuration and the operating range of the forces of each thruster, training data is generated. Then a model incorporating the above-mentioned constraints is trained offline according to the loss functions described in Section III-B. A significant feature of such a training regime is that the DNN does not require real data from ship maneuvering operations [11].

A related work from the authors is published in [12], in which a one layer feedforward neural network was applied for control allocation. The current paper has the following novelties:

- The model used for allocation consists of recurrent layers, which facilitates constraining rates without feeding thruster commands from a real operation to the network (see Section III-B3).
- Implementation of constraints such as power minimization and sector constraints is made possible through the use of the Autoencoder-like structure of the network (see Section II-B)

II. Background

A. Generalized Inverse Control allocation

The Generalized Inverse (GI) method given in [2] provides for an unconstrained solution of the control allocation. The expression for the generalized force, $\boldsymbol{\tau}$, is given in (1). $\boldsymbol{\tau}$ is produced jointly by the thrusters and holds the longitudinal/lateral forces and moment about the vertical axis of the ship.

$$\begin{aligned}\boldsymbol{\tau} &= \begin{bmatrix} 0 & c(\alpha_2) & c(\alpha_3) \\ 1 & s(\alpha_2) & s(\alpha_3) \\ l_2 & l_1 s(\alpha_2) & l_1 s(\alpha_3) \\ & -l_3 c(\alpha_2) & -l_4 c(\alpha_3) \end{bmatrix} \times \begin{bmatrix} F_1 \\ F_2 \\ F_3 \end{bmatrix} \\ &= \boldsymbol{B}(\boldsymbol{\alpha})\boldsymbol{u}_f \end{aligned} \tag{1}$$

where $\boldsymbol{B}$ is the thruster configuration matrix, $\boldsymbol{\alpha}$ contains azimuth angle commands, $\boldsymbol{u}_f$ contains thruster force commands and $s(\cdot)$ and $c(\cdot)$ represent the sine and cosine functions. The remaining variables of (1) are given in Fig. 1(b). If the thrusters are non-rotatable, (1) may be solved using the Moore-Penrose pseudo-inverse of $\boldsymbol{B}$. This yields the optimal command vector in (2), not accounting the potential in-feasibility of the commands.

$$\boldsymbol{u}_f = \boldsymbol{B}^{\dagger}(\boldsymbol{\alpha}_0)\boldsymbol{\tau} \tag{2}$$

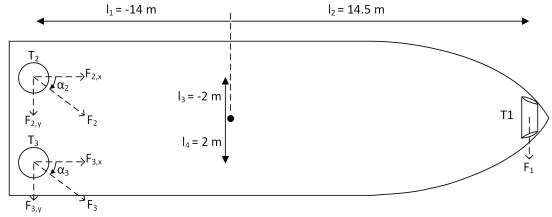
This method will be used in the simulation example of this study with a set of fixed angles, $\boldsymbol{\alpha}_0$. According to Fig. 1(b), two thruster types appear in the vessel used in this study: One variable-speed tunnel thruster (T_1) and two identical, variable speed, azimuth thrusters (T_2 and T_3). The simulation models used in this study are based on the Research Vessel (R/V) Gunnerus [13] and its propulsion system [14]. A view of the real ship is given in Fig. 1(a), and its physical parameters are given in Table I.

Force commands are issued from the allocator to a Proportional-Integral (PI) controller per thruster. The PI controller outputs a torque reference to an electrical motor based on the error between the estimated thrust and the reference thrust from the allocator. The motor then outputs an RPM to the hydrodynamic model of the thruster, which outputs thrust. In Fig. 4, three PI-motor-thruster subsystems are contained in the *Thrusters* block. Each thruster exerts a 3-dimensional force at its location on the ship hull. The combined thruster force is then applied, along with the additional excitation forces, in the time-domain ship simulator VeSim [13] (see the *Ship* block of Fig. 4).

TABLE I
PARAMETERS OF THE VESSEL AND THRUSTERS USED IN THE SIMULATION EXPERIMENT.

Parameter	Value
Deadweight (t)	107
Length between perpendiculars (m)	28.9
Breadth middle (m)	9.6
Draught (m)	2.8
$T_1/T_2/T_3$ max N/s	1000
T_2/T_3 max °/s	10
T_1 max N	±30000
T_2/T_3 max N	-30000,60000
T_2/T_3 max °	±180

(a) Starboard view of the R/V Gunnerus



(b) Thruster layout of the vessel used in this study

Fig. 1. A virtual model of the RV Gunnerus was applied in this study.

B. DNN

The neural network architecture used in this paper resembles an Autoencoder network. However, Autoencoder networks typically aim to reduce the dimension of the latent space, also known as the Code, to represent the input space samples using a more compact representation (ie. compression of information) [15]. In this work the structure of the Autoencoder network is exploited to facilitate training and enforce constraints such that the encoder-part of the network in Fig. 2 acts as a control allocator, similar to the work of Huan et al. [11]. We expand this approach in the direction of the maritime domain and also facilitate for constraints in terms of the rate change of the azimuth thruster angles. Section III describes the dimensions of the input, latent and output spaces.

The two parts of the network; the *Encoder* and the *Decoder* were assigned three layers each based on a manual search considering training/execution time versus performance. Hyperparameters were also manually selected based on the performance of the model when applying a separate validation dataset. 64 Long Short-Term Memory (LSTM) nodes were applied in the first two layers of each part of the network. A regular densely connected layer made up the final layer for both parts of the network, with a dimension of 5 for the *Encoder* and 3 for the *Decoder*. LSTM nodes were selected based on their ability to retain memory across input sequences [16]. This functionality allows for applying rate constraints, where there is a need to store temporal information.

III. Neural Network-based Control Allocation

The neural network allocator used in this study is in the form of a DNN. Fig. 2 shows the architecture of the network, which interfaces with the motion controller through the force $\boldsymbol{\tau}$ and outputs individual thruster commands through $\hat{\boldsymbol{u}}$. The latter contains force and angle commands while the former contains the force request components along the three dimensions of freedom for the vessel: surge, sway and yaw.

A. Data

The data used for training the DNN is artificially generated based on a user-defined force and angle command range for $\boldsymbol{u} = [F_1, F_2, \alpha_2, F_3, \alpha_3]^T$ shown in (3). Information about the thruster locations is also used to transform a set of force/angle commands into generalized forces, which is what the allocator receives from the motion controller. In that way, inputs and targets for the supervised training procedure are calculated according to (1), while the entries of $\boldsymbol{u}$ are used in the calculation of losses during training (see Section III-B). Dynamics in the control sequences, $\boldsymbol{\tau}$, are not considered in the construction of the training data. It changes only due to the prescribed force/angle commands. For the azimuth thrusters a range must be defined for both the force and the azimuth angle, while the tunnel thruster requires only a force range. There exists a trade-off between the resolution within each range and the corresponding training performance, which hinges on the generalization ability of the DNN. The ranges shown in (3) are based on that trade-off, as well as the size of the resulting dataset and the required range in which the vessel will be operating. The required range of commands depends on the aggressiveness of the motion controller and the maneuvering regime.

An important consideration at this stage is that the ranges observed in the training data must exceed the constraints applied during training. If a loss function always returns zero, it has no impact on the network training. This applies to the losses described in Sections III-B2, III-B3 and III-B5 as these are only active if the constraints are violated. The simulation examples in this study require the ship to perform low-speed maneuvering and stationkeeping with moderate wind speeds. This results in low force magnitudes relative to the corresponding maximum force magnitudes of each thruster given in Table I.

$$\begin{aligned} F_1 &\in [-10000, 10000]N \\ F_2 &\in [-5000, 5000]N \\ \alpha_2 &\in [-180, 180]^\circ \\ F_3 &\in [-5000, 5000]N \\ \alpha_3 &\in [-180, 180]^\circ \end{aligned} \tag{3}$$

F_1 is the force of T_1 , F_2 and α_2 are the force and thruster angle of T_2, and F_3 and α_3 are the force and thruster angle of T_3. Drawing one million samples from these ranges, using a uniform randomized selection strategy, yields the dataset for the commands, $\boldsymbol{u}$. It has a dimension of 5 x 1000000. Applying 90% of the columns in $\boldsymbol{u}$ to (1) results in the training dataset, $\boldsymbol{\tau}$, which holds the generalized force. A standardization scaling procedure is performed on the training dataset to facilitate proper training of the DNN.

B. Training

The training operation is performed iteratively, where each iteration involves passing every training data sample to the network, known as an epoch. Each epoch contains a fixed number of subsets of the training dataset (mini-batches, denoted by the superscript i), and weight updates are carried out after each mini-batch has been processed. In this study a mini-batch size of 1024 was chosen, resulting in roughly 880 mini-batches. If, for three consecutive epochs, the combined loss (see Section III-B6) evaluated on a separate validation dataset does not improve, the training procedure is terminated.

In order to fulfill constraints in terms of meeting the motion controller force requests, keeping the magnitude and rate of the control vector within certain bounds, and facilitating power minimization, loss functions were designed. The output of the loss functions in Sections III-B2-III-B5 determine the weight updates to each layer in the Encoder-part of Fig. 2 during the backpropagation training scheme. The loss function in Section III-B1 applies to the weight update of all layers in the DNN. Loss functions serve the same purpose as objective functions of optimization-based allocation methods. Note that the entire training procedure should be performed offline. In the following $f_e(\boldsymbol{\tau})$ represents the Encoder-part of the network, $f_d(\hat{\boldsymbol{u}})$ represents the Decoder-part of the network while $f(\boldsymbol{\tau})$ is the overall model. For ease of notation all variables included in this section are connected to the DNN training process.

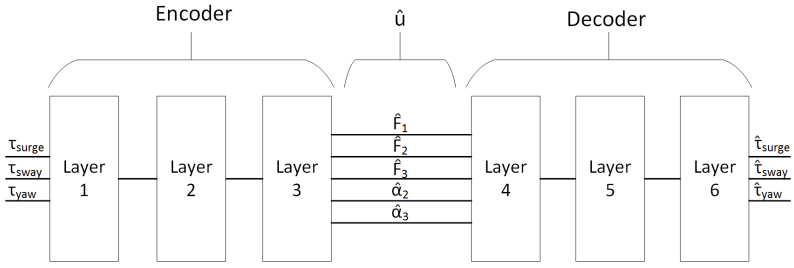


Fig. 2. Structure of the neural network used for modelling the control allocation problem.The input of Layer 1 is the force commanded by the motion controller while the output of Layer 3 is the commands issued to each thruster.

A key feature of this training regime is a loss function to ensure that the output of Layer 3 is expanded. Without this feature the neural allocator would not have any incentive to utilize the entirety of the feasible command region, since the weights of Layer 4-6 would absorb the training error. By re-scaling the output of Layer 3 according to the ranges seen in (3), calculating the resulting estimated generalized force and comparing it to the ground truth, the commands are forced into ranges that minimize the requested generalized force (see (5)). First the expression for the generalized force estimate (calculated from the estimated commands $\hat{\boldsymbol{u}}_f(k) = [\hat{F}_1(k), \hat{F}_2(k), \hat{F}_3(k)]^T$ and $\hat{\boldsymbol{\alpha}}(k) = [\hat{\alpha}_2(k), \hat{\alpha}_3(k)]^T$) at mini-batch index $k \in [0, 1023]$ is given in (4).

$$\hat{\boldsymbol{y}}_{cmd}(k) = \boldsymbol{B}(\hat{\boldsymbol{\alpha}}(k))\hat{\boldsymbol{u}}_f(k) \tag{4}$$

The result of (4) is a 3 x 1024 matrix of estimated surge/sway force and moment about the yaw axis of the ship. $\hat{\boldsymbol{y}}_{cmd}$ is applied in (5) to get the mean squared error (MSE) loss relative to the ground truth $\boldsymbol{y}_j$.

$$L_0 = \sum_{j=0}^{2} (\boldsymbol{y}_j^i - \hat{\boldsymbol{y}}_{j,cmd}^i)^2 \tag{5}$$

This loss function does not contribute towards satisfying the constraints imposed on the neural allocator in the training process.

1) Minimize error in allocated force: The primary goal of any control allocation method is to minimize the difference between the motion controller force request and the combined generalized force produced by all thrusters. This goal is enforced through a loss function applied to the output layer of the DNN (Layer 6) seen in (6).

$$L_1 = \sum_{j=0}^{2} (\boldsymbol{y}_j^i - \hat{\boldsymbol{y}}_j^i)^2 \tag{6}$$

$\boldsymbol{y}^i$ represents the ground truth for the three-dimensional generalized force, while $\hat{\boldsymbol{y}}^i = \hat{\boldsymbol{\tau}}^i = f(\boldsymbol{\tau}^i)$ is the corresponding estimate made by the final layer of the complete model. (6) represents the standard MSE loss which is often used in NN training. During training this loss quantifies how well the output of the network matches/reconstructs the generalized force. It offers a way to shape the weights of the DNN towards minimizing the error in allocated generalized force. However, it does not contribute to keep the secondary constraints of the allocator. This will be covered in the following sections.

2) Thruster command magnitude: Each thruster has a finite range of operation in terms of force (for thrusters T_1, T_2, T_3) and azimuth angles (for thrusters T_2 and T_3). From Table I the maximum force of T_2 and T_3 are not symmetrical about zero. However, for simplicity and the limited force range required in the case study, a symmetrical maximum force constraint is applied according to (7)

$$L_2 = \sum_{l=0}^{4} \max((|\hat{\boldsymbol{u}}_l^i| - \boldsymbol{u}_{l,max}), 0) \tag{7}$$

L_2 penalizes commands exceeding the threshold set for each command in $\boldsymbol{u}$. The $max()$-function and the $|.|$-function perform element-wise operations for all five variables in $\boldsymbol{u}^i$, denoted by the subscripted variable l. Note that this results in a soft threshold for the command output and, depending on the relative weighting of the loss functions, an excursion beyond the maximum limit may occur. The same is true for the constraints described in Section III-B3 and III-B5. A higher relative weighting is therefore applied to these constraints to keep within the specified constraint thresholds, yielding more conservative allocated commands.

3) Rate changes: For the network to learn to penalize large rate changes, the output of the Encoder-part of the network, $\hat{\boldsymbol{u}}^i = f_e(\boldsymbol{\tau}^i)$, is compared against its copy that has been shifted along the second dimension by one index. This results in a matrix containing the predicted rates of each value in $\hat{\boldsymbol{u}}^i$ for all 1024 samples in the mini-batch. If the rates exceed a certain value (see Table I), a loss is incurred according to (8).

$$L_3 = \sum_{l=0}^{4} max(|\hat{\boldsymbol{u}}_l^i - shift(\hat{\boldsymbol{u}}_l^i)| - \Delta \boldsymbol{u}_{l,max}, 0) \tag{8}$$

4) Power consumption minimization: Without explicitly telling the network to minimize power consumption, ie. penalize the use of thrust, arbitrary values within the allowable ranges of the constraints mentioned in Sections III-B2 and III-B3 may occur. Therefore, using the fact that the thrust is a quadratic function of the RPM and the consumed power is a cubic function of the RPM [3], the loss function L_4 in (9)

is applied. The power required to rotate the azimuth thrusters is not considered.

$$L_4 = |\hat{\boldsymbol{u}}_0^i|^{(3/2)} + |\hat{\boldsymbol{u}}_1^i|^{(3/2)} + |\hat{\boldsymbol{u}}_3^i|^{(3/2)} \tag{9}$$

5) Azimuth sectors: Constraining the azimuth angles α_2 and α_3 is beneficial in terms of avoiding a decrease in thruster efficiency due to disturbance of inflow velocities by neighbouring thrusters [17]. Two disallowed sectors are defined, since thrusters T_2 and T_3 may produce both positive and negative thrust. They are: $\underline{\alpha}_c = [\underline{\alpha}_{0,c}, \underline{\alpha}_{1,c}] = [-100, -80]$ deg and $\overline{\alpha}_c = [\overline{\alpha}_{0,c}, \overline{\alpha}_{1,c}] = [80, 100]$ deg. The disallowed sectors are visualized in Fig. 3.

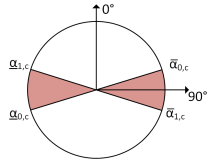


Fig. 3. Disallowed sectors of thrusters T_2 and T_3 are given in red.

When training the network to avoid these sectors the loss functions in (10) and (11) are applied.

$$\underline{L}_5 = \sum_{l\in[2,4]} (\hat{\boldsymbol{u}}_l^i < \underline{\alpha}_{1,c}) \times (\hat{\boldsymbol{u}}_l^i > \underline{\alpha}_{0,c}) \tag{10}$$

$$\overline{L}_5 = \sum_{l\in[2,4]} (\hat{\boldsymbol{u}}_l^i < \overline{\alpha}_{1,c}) \times (\hat{\boldsymbol{u}}_l^i > \overline{\alpha}_{0,c}) \tag{11}$$

The index $l \in [2, 4]$ denotes the azimuth angle of T_2 and T_3, respectively. Note that the $\times$-operator and $<, >$-operators perform element-wise operations on $\hat{\boldsymbol{u}}_l^i$, resulting in loss values for each sample in a mini-batch. The $<, >$-operators yield a value of 1 if violating the sector constraint, otherwise zero. The overall loss for the two disallowed sectors for T_2 and T_3 is given by $L_5 = \underline{L}_5 + \overline{L}_5$.

6) Combined loss: A combination of the aforementioned loss functions is used to meet the constraints given in Section III-B. The combined loss $L = k_0L_0 + k_1L_1 + k_2L_2 + k_3L_3 + k_4L_4 + k_5L_5$ is applied to the network optimizer, which performs the adaptation of the weights in $f_e()$ (due to L) and $f(\boldsymbol{\tau})$ (due to k_1L_1). The scaling factors $k_0 - k_5$ weight the importance of each loss such that the performance may be altered depending on user requirements. Re-scaling the output of $f_e(\boldsymbol{\tau})$ in the training procedure calls for small scaling factors in the weighted sum for L. Table II gives values for the case study presented in Section IV.

C. Allocation

Once the DNN is trained and validated offline, it may be used to obtain the commands for each thruster. This is done by performing a forward pass over $f_e()$ using the force request from the motion controller as input. The execution time is fixed and depends on the complexity of the neural allocator (number of layers, type of layers, number of nodes in each layer etc.) and the hardware it runs on.

In the event of thruster failure situations, operations that require thrust to be produced in certain directions only, or if operational considerations require less penalization of power consumption, re-training the neural allocator is required. For time-critical events, a set of pre-trained neural allocators may be stored and used on demand (eg. if T_2 fails, a neural allocator disregarding this thruster may be applied). Otherwise, the training procedure may be performed online and applied when ready. Using a low-rate graphics processing unit, the training time for the neural allocator used in this study averaged at roughly two minutes.

IV. Case studies and results

The procedure for constrained control allocation using a DNN, described in Section III-B, was tested in a simulator. The physical objects included in the simulation were modelled in separate containers, known as functional mock-up units. So were the software required for motion control and allocation. Fig. 4 shows the components of the simulation and how they interact. The motion controller was based on the PID controller described in [18].

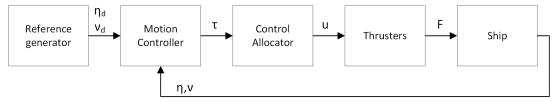


Fig. 4. Simulation components.

A. Low-speed maneuvering

To create smooth references for the controller during a low-speed maneuvering test, a reference generator was applied, which output both position/heading references and their corresponding velocity references. A 4-corner DP test was constructed to gauge the performance of the neural allocator described in Section III [19]. The outline of the procedure is given below, where the letters of the itemized list correspond to letters in Fig. 6:

- A: Initiate the vessel at a heading of 0 degrees at location (0 m north, 0 m east).
- B: Move 20 m straight north. Start time: 0 s.
- C: Move 20 meters straight west. Start time: 250 s.
- D: While at location (20 m north, 20 m east), rotate the vessel to achieve a heading of 315 degrees. Start time: 450 s.
- E: While maintaining a heading of 315 degrees, move 20 m south. Start time: 650 s.
- F: Move 20 m east while also rotating to a heading of 0 degrees. Start time: 850 s.

1) Results: The GI allocator serves as a baseline for comparison relative to the neural allocator proposed in this study. Three candidate configurations for the fixed thruster angles were tested (see Fig. 5). Based on mean power consumption and path errors incurred while maneuvering through the test procedure given in the previous section, the fixed angles $\alpha_2 = -45$ and $\alpha_3 = 45$ were selected. Note that for fixed

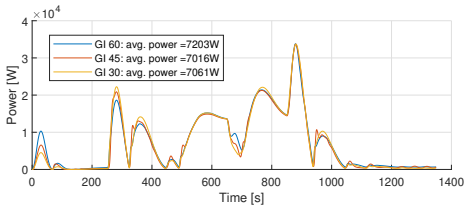


Fig. 5. Power consumption in Watts for three configurations of the fixed-angle GI allocator.

angles the GI-allocated control forces are optimal in terms of energy [2]. However, the fixed 45 degree azimuth angles offers sub-optimal thrust directions for the translations and rotations performed in the 4-corner maneuvering test.

Fig. 6 shows the simulated vessel path for the GI fixed-angle (± 45 degrees) allocator and the neural allocator. The simulation was run for both allocators using identical simulation parameters (path references, control parameters etc.). A more detailed comparison of the maneuvering performance, using each of the two allocation implementations, is given in Fig. 7. Thruster commands issued by the neural allocator and the GI allocator are shown in Fig. 8. The GI force commands show somewhat higher maximum values. This is due to being restricted to fixed azimuth angles for thrusters T_2 and T_3. The neural allocator absorbs some of the variation in RPM by changing angles of thrusters T_2 and T_3.

Fig. 9 shows the resulting losses observed during the 4-corner test for the neural allocator. Notably, the losses L_2, L_3 and L_5 are not included since they remain zero if constraints are not breached. They contribute during the training phase. In Fig. 8 the azimuth angles do not cross the $\pm$ 80 deg constraints and the maximum forces are below the constraint set during training of $\pm$ 8 kN for T_1 and $\pm$ 4 kN for T_2/T_3. Due to maneuvering at low speed the force commands issued

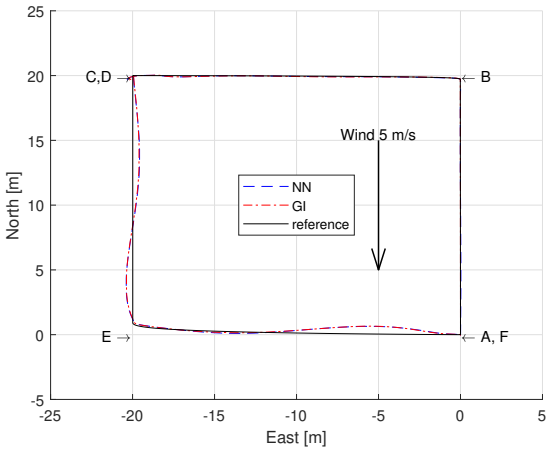


Fig. 6. Path taken for both the neural allocator and the GI allocator. The latter apply fixed azimuth angles of $\alpha_2 = -45$ and $\alpha_3 = 45$ degrees.

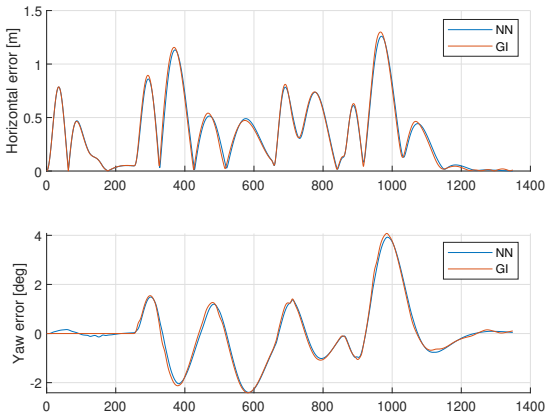


Fig. 7. Comparison of the positioning performance of the GI allocator and the neural allocator.

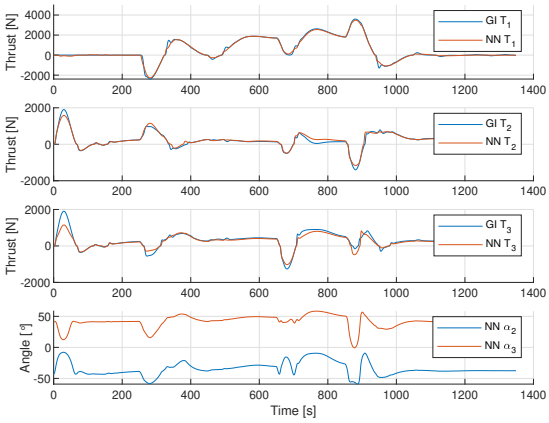


Fig. 8. Force and azimuth angle commands issued by both allocators. Note that the bottom plot only contains the azimuth angles of the neural network allocator.

by the GI allocator resulted in force rates within 20 % of the maximum values given in Table I. This allowed the GI allocator to match the generalized force requests perfectly. Similar force rates are seen for the neural allocator, while the observed azimuth angle rate kept within 70 % of the $10^\circ/s$ limit. The similar trends of the bottom two plots of Fig. 9 indicate that the overall loss, L, is dominated by the power loss, L_4. This suggests that the training procedure has been successful in shaping the network, such that the autoencoder estimates the input well (low L_1 value). The relatively low L_0 value indicates that the generalized force, achieved through multiplying the allocated commands with the thruster configuration matrix, closely matches the requested generalized force.

As the training procedure of the neural allocator applies several loss functions (to keep within the constraints mentioned in Section III-B), the relative weighting between them dictates

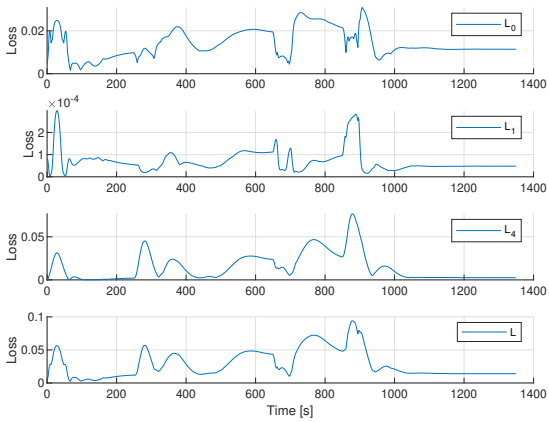


Fig. 9. Losses incurred by the neural allocator during the 4-corner test.

the allocation performance. In this case, the values of Table II were used to scale the individual loss values L_0-L_5.

TABLE II
THE VALUES OF THE LOSS SCALING FACTORS k_0-k_5.

Parameter	Value
k_0	10^0
k_1	10^0
k_2	10^{-1}
k_3	10^{-7}
k_4	10^{-7}
k_5	10^{-1}

The average power consumption for the two allocator instances is similar, with an average power decrease of 3.6 % using the neural allocator (see Fig. 10). The largest power reduction is seen in the period t=0-100 s, which is when the ship performs a straight north move. In this period the azimuth angles for T_2 and T_3 are closer to zero which leads to a more efficient use of thrust (see Fig. 8). Constraining the

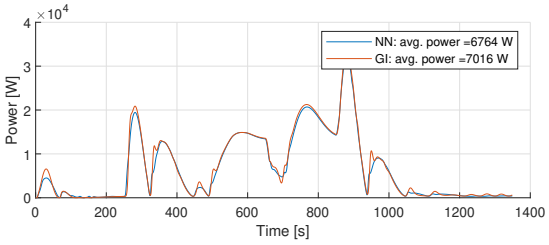


Fig. 10. Power consumption of the GI allocator and the neural allocator.

power consumption effectively led to more variation in the azimuth angles of thrusters T_2 and T_3. In allowing less use of thrust, the thrusters are forced to find more effective angles to operate. The user must therefore weight the importance of power minimization versus the advantage of having low azimuth angle rate changes. At a certain point, constraining power also influences on the ability of the allocator to produce the requested generalized force.

In terms of computational speed the present approach takes 6 ms per execution. To put this in context, the execution time of the MPC algorithm described in [9] was reported to be approximately 10 ms. They also compared against the Sequential Quadratic Programming (SQP) approach of [3] and found that the average computational time was 0.385 ms, while the original study reported a run time of 2 ms. Advances in computational power could explain this difference. The neural allocator is therefore 1.6 times faster than the MPC approach and 15.6 times slower than the SQP approach.

B. Impact of scaling power loss

The effect of varying the scaling of the loss described in Section III-B4 is investigated in this section through a stationkeeping simulation test. The implementation of this loss yields a positive value for all time if the force produced by one of the thrusters is greater than zero. An increase in scaling factor k_4, relative to the other scaling factors, implies that the trained network favors lower thrust commands. Scaling factors other than k_4 remains equal to that of Table II in this test.

Stationkeeping involves the use of thrusters to keep the ship at a fixed location and heading [20]. A constant external disturbance was applied in the form of a uniform wind field coming from the east at a magnitude of 10 m/s. By allowing the wind direction to change quickly (from east to northeast in five seconds), the effect of varying k_4 will be examined.

Fig. 11 shows the total loss, L, and the overall power consumption. The sensitivity of L with respect to changes in k_4 is evident from the substantial increase in L for increasing k_4. Increasing k_4 also causes azimuth angles to be used more efficiently. This suggests an indirect coupling between L_4 and the azimuth angle commands produced by the allocator, since only the thrust is penalized in L_4. The top plot of Fig. 12 shows the azimuth angles of T_3, for 3 and 5 times the original k_4, being closer to the 80 degree constraint. This means that it more efficiently counters the wind disturbance coming from the east. The coupling effect between scaling factors increases the difficulty of finding suitable parameters for k_0-k_5.

V. CONCLUSION

The proposed neural network allocator provides similar functionality as optimization-based methods by accommodating constraints in its training phase. Also, a comparable performance was found relative to the GI implementation in terms of matching the force request from the motion controller, power consumption and ability to perform low-speed maneuvering. Modelling the neural allocator using only knowledge of the thruster configuration, the user-specified constraints and loss functions with scaling, allows for pre-training of the neural allocator without the need for data from a real operation. A computation time of 6 ms was observed, which enables real-time operation.

The proposed neural network allocator does not enforce constraints in a strict way. This means that, depending on how well the training process is posed, constraints may be

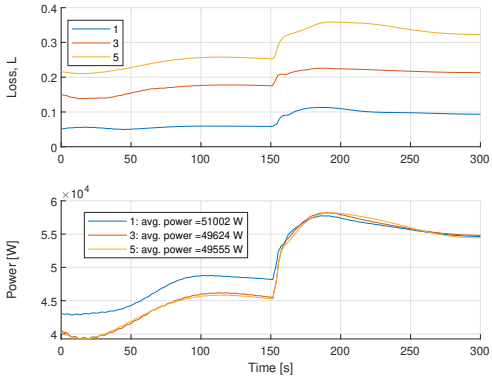


Fig. 11. Power consumption (bottom) and overall loss (top) observed during the stationkeeping test. The legend entries indicate: 1: $k_4 = 1 \times 10^{-7}$, 3: $k_4 = 3 \times 10^{-7}$, 5: $k_4 = 5 \times 10^{-7}$.

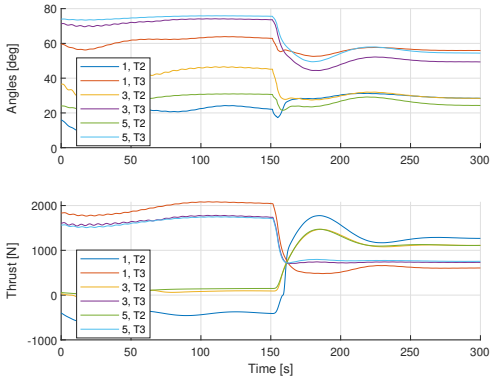


Fig. 12. Angles (bottom) and thrust (top) for thrusters observed during the stationkeeping test. The legend entries indicate: 1: $k_4 = 1 \times 10^{-7}$, 3: $k_4 = 3 \times 10^{-7}$, 5: $k_4 = 5 \times 10^{-7}$. T_2 and T_3 indicate the azimuth thrusters.

breached. For instance, if too little emphasis is put on penalizing azimuth angle rates, a rapid change in the motion controller output might lead to commanding unattainable azimuth angles. Means of enforcing hard constraints will hence be further investigated.

In case of failures, which implies that the neural allocator needs to be re-trained, the training time of the allocator is critical. In future developments of this work, this will be a main focus. We also aim to implement this research, along with efforts from aligned research activity at NTNU Aalesund, on the R/V Gunnerus to realize an autonomous ship operation.

VI. Acknowledgement

This work was supported in part by a grant from the Knowledge-Building Project for Industry "Digital Twins for Vessel Life Cycle Service" (Project 280703) and in part by a grant from the Research-Based Innovation "SFI Marine Operation in Virtual Environment" (Project 237929) in Norway. The third author was partially funded by the Norwegian Research Council (NTNU AMOS) at the Norwegian University of Science and Technology (grant no. 223254).

References

[1] T. A. Johansen and T. I. Fossen, "Control allocation - A survey," *Automatica*, vol. 49, no. 5, pp. 1087–1103, 2013.

[2] T. I. Fossen, *Handbook of Marine Craft Hydrodynamics and Motion Control*. John Wiley and Sons Ltd., 2011.

[3] T. A. Johansen, T. I. Fossen, and S. P. Berge, "Constrained Nonlinear Control Allocation With Singularity Avoidance Using Sequential Quadratic Programming," *IEEE Trans. Control Syst. Technol.*, vol. 12, no. 1, pp. 211–216, 2004.

[4] O. J. Sørdalen, "Optimal thrust allocation for marine vessels," *Control Eng. Pract.*, vol. 5, no. 9, pp. 1223–1231, 1997.

[5] A. Veksler, T. A. Johansen, R. Skjetne, and E. Mathiesen, "Thrust Allocation With Dynamic Power Consumption Modulation for Diesel-Electric Ships," *IEEE Trans. Control Syst. Technol.*, vol. 24, no. 2, pp. 578–593, 2016.

[6] M. Rindaroey and T. A. Johansen, "Fuel optimal thrust allocation in dynamic positioning," in *IFAC Conf. Control Appl. Mar. Syst.* IFAC, 2013, pp. 43–48.

[7] T. A. Johansen, T. P. Fuglseth, P. Tøndel, and T. I. Fossen, "Optimal constrained control allocation in marine surface vessels with rudders," *Control Eng. Pract.*, vol. 16, no. 4, pp. 457–464, 2008.

[8] F. Arditti, F. L. Souza, T. C. Martins, and E. A. Tannuri, "Thrust allocation algorithm with efficiency function dependent on the azimuth angle of the actuators," *Ocean Eng.*, vol. 105, pp. 206–216, 2015.

[9] S. Skjong and E. Pedersen, "Nonangular MPC-Based Thrust Allocation Algorithm for Marine Vessels - A Study of Optimal Thruster Commands," *IEEE Trans. Transp. Electrif.*, vol. 3, no. 3, pp. 792–807, 2017.

[10] M. Chen, "Constrained Control Allocation for Overactuated Aircraft Using a Neurodynamic Model," *IEEE Trans. Syst. Man, Cybern. Syst.*, vol. 46, no. 12, pp. 1630–1641, 2016.

[11] H. Huan, W. Wan, C. We, and Y. He, "Constrained Nonlinear Control Allocation based on Deep Auto-Encoder Neural Networks," *2018 Eur. Control Conf. ECC 2018*, pp. 2081–2088, 2018.

[12] R. Skulstad, G. Li, H. Zhang, and T. I. Fossen, "A Neural Network Approach to Control Allocation of Ships for Dynamic Positioning," *IFAC-PapersOnLine*, vol. 51, no. 29, pp. 128–133, 2018.

[13] V. Hassani, A. Ross, Ø. Selvik, D. Fathi, F. Sprenger, and T. E. Berg, "Time Domain Simulation Model For Research Vessel Gunnerus," in *Int. Conf. Ocean. Offshore Arct. Eng.*, 2015, pp. 1–6.

[14] S. Skjong, M. Rindarøy, L. T. Kyllingstad, V. Æsøy, and E. Pedersen, "Virtual prototyping of maritime systems and operations: applications of distributed co-simulations," *J. Mar. Sci. Technol.*, vol. 23, no. 4, pp. 835–853, 2018. [Online]. Available: http://dx.doi.org/10.1007/s00773-017-0514-2

[15] W. Liu, Z. Wang, X. Liu, N. Zeng, Y. Liu, and F. E. Alsaadi, "A survey of deep neural network architectures and their applications," *Neurocomputing*, vol. 234, no. November 2016, pp. 11–26, 2017.

[16] K. Greff, R. K. Srivastava, J. Koutník, B. R. Steunebrink, and J. Schmidhuber, "LSTM: A Search Space Odyssey," *IEEE Trans. Neural Networks Learn. Syst.*, vol. 28, no. 10, pp. 2222–2232, 2017.

[17] P. Yadav, R. Kumar, S. K. Panda, and C. S. Chang, "Optimal thrust allocation for semisubmersible oil rig platforms using improved harmony search algorithm," *IEEE J. Ocean. Eng.*, vol. 39, no. 3, pp. 526–539, 2014.

[18] A. J. Sørensen, "A survey of dynamic positioning control systems," *Annu. Rev. Control*, vol. 35, no. 1, pp. 123–136, 2011.

[19] S. A. T. Værnø and A. H. Brodtkorb, "AMOS DP Research Cruise 2016: Academic Full-Scale Testing of Experimental Dnamic Positioning Control Algorithms Onboard R/V Gunnerus," in *Proc. ASME 2017 36th Int. Conf. Ocean. Offshore Arct. Eng.*, 2017, pp. 1–10.

[20] 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 Robot. Autom. Mag.*, vol. 26, no. 3, pp. 39–51, 2019.

Robert Skulstad, Intelligent Systems Lab, Department of Ocean Operations and Civil Engineering, Norwegian Univer-

sity of Science and Technology (NTNU) Aalesund. E-mail: robert.skulstad@ntnu.no.

Guoyuan Li, Intelligent Systems Lab, Department of Ocean Operations and Civil Engineering, NTNU Aalesund. E-mail: guoyuan.li@ntnu.no.

Thor I. Fossen, Department of Engineering Cybernetics, NTNU Trondheim. E-mail: thor.fossen@ntnu.no.

Bjørnar Vik, Kongsberg Maritime, Aalesund. E-mail: bjornar.vik@km.kongsberg.com.

Houxiang Zhang, Intelligent Systems Lab, Department of Ocean Operations and Civil Engineering, NTNU Aalesund. E-mail: hozh@ntnu.no.

ISBN 978-82-326-6093-3 (printed ver.)
ISBN 978-82-326-6605-8 (electronic ver.)
ISSN 1503-8181 (printed ver.)
ISSN 2703-8084 (online ver.)

NTNU
Norwegian University of
Science and Technology