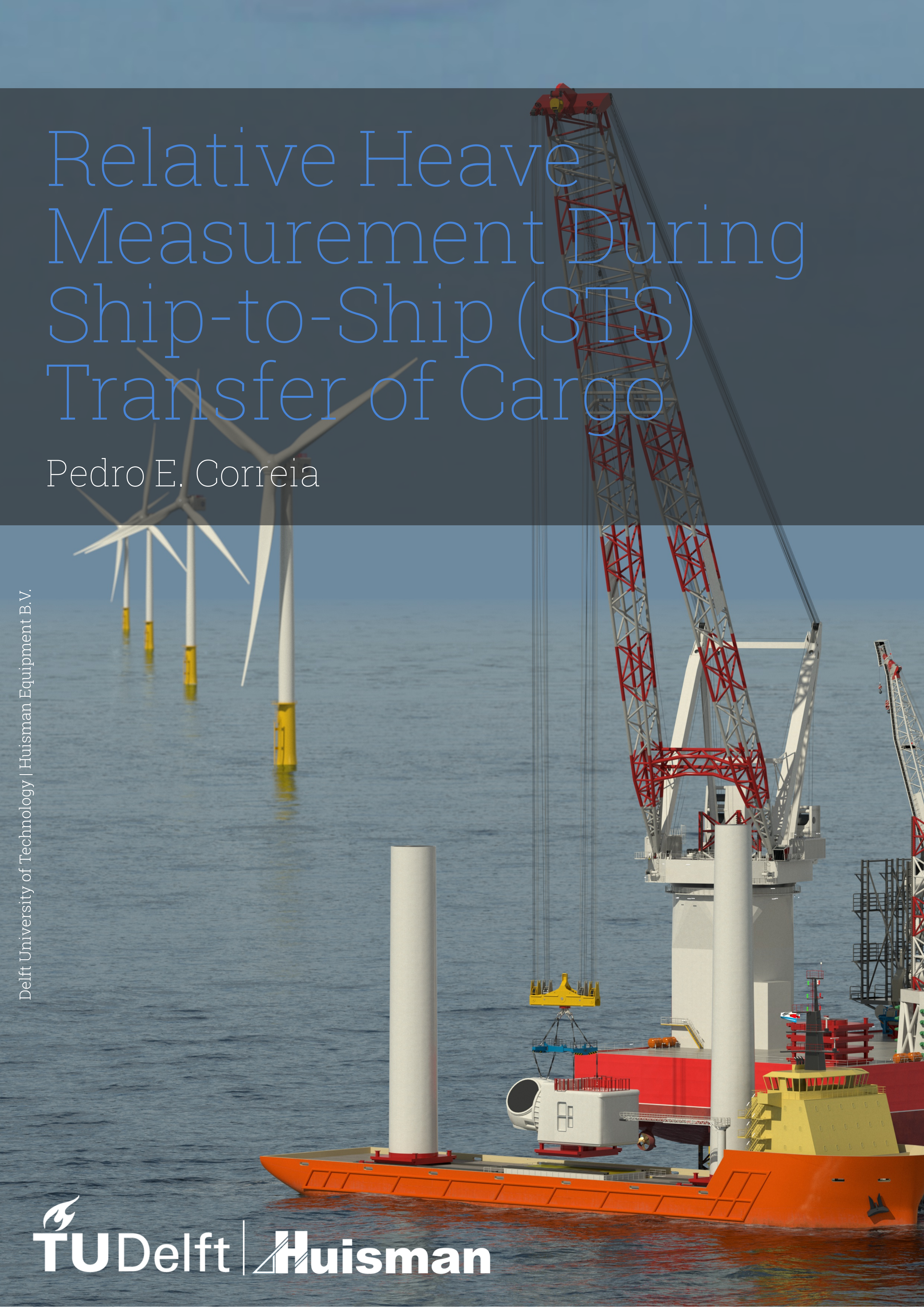
Relative Heave Measurement During Ship-to-Ship (STS) Transfer of Cargo

Pedro E. Correia

Delft University of Technology | Huisman Equipment B.V.



Relative Heave Measurement During Ship-to-Ship (STS) Transfer of Cargo

by

Pedro E. Correia

in partial fulfilment of the requirements for the degree of

Master of Science
in Mechanical Engineering

at the Department of Maritime and Transport Technology of the Faculty Mechanical, Maritime and Materials Engineering of Delft University of Technology
to be defended publicly on Tuesday, December 15th, 2023, at 10:00 AM

Student number:	5540143
MSc track:	Multi-Machine Engineering
Report number:	2023.MME.8883

Thesis supervisor:	Dr. V. Reppa	TU Delft supervisor, 3mE
Thesis committee:	Dr. Y. Pang	TU Delft committee Chair, 3mE
	Dr. A. Grammatikopoulos	TU Delft committee member, 3mE
	Dr. Holger Caesar	TU Delft committee member, 3mE
	Ir. J. Dijkstra	Company supervisor, Huisman Equipment B.V.

Date: 7th December, 2023

Cover image: Ship-to-ship cargo transfer [1]

An electronic version of this thesis is available at http://repository.tudelft.nl/.

Preface

This thesis marks the culmination of hard work, passion, and an unwavering desire to solve every problem encountered along the way. The intriguing topic of Relative Heave Compensation (RHC) and its practical relevance has motivated me to move forward with the project, aiming to make a significant contribution not only to academia but also to the industry, with the hope that my work would have a tangible impact within the field.

I hope this thesis is easily readable, regardless of your background, and that through it, you can experience the same journey I undertook to achieve my research goals. Furthermore, I hope this research will serve as an essential guide for future studies and practical implementation.

Reaching the end of this project has made me reflect on all my experiences over the past year, leaving me with a bittersweet feeling as I conclude my studies at TUDelft. The good and less good experiences shared with those around me, directly or indirectly, have been essential for both my personal and professional growth.

On this note, I extend my gratitude to all those who joined me on this journey. I begin by thanking my TUDelft supervisors, Vasso and Yusong, who consistently provided me with critical and scientific feedback on my work. Warm thanks to everyone at Huisman who welcomed me greatly. Especially my company supervisors, Robert for always being ready to help and test something out; Jeroen, whose experience guided me and pushed me to excel in this project; and Jesse, whose kindness and extensive knowledge were invaluable during the various phases of my project.

Beyond this project, I want to express my appreciation to all my TUDelft colleagues, now friends, too numerous to name, who shared memorable moments with me. Similarly, my Portuguese friends, also countless in number, who became my extended family away from home.

Living abroad makes those you share your time with the closest thing to a family. I consider myself fortunate to have lived alongside remarkable individuals — individuals whose kindness and intelligence are immeasurable. They not only provided unwavering support during challenging times but also contributed to making some moments the absolute highlights of my master's journey. My heartfelt thanks go to Andrea, Bernardo, Miguel, and Diogo. They occupy a special place in my heart, and I extend an open invitation for them to reach out whenever needed.

Lastly, thank you to my family, especially my mom and dad, who made me who I am today. Their love and immense patience have been the cornerstone of all my achievements. They have consistently been my biggest supporters, allowing me to explore and embrace new experiences, knowing I can always fall back on them. My endless wish is to make them proud for all they have given me.

With these acknowledgements, I leave you with a thought that has echoed throughout my thesis: ”I wish I could skip the part where I am unsure of what I am doing and go to the one where I know exactly what I am supposed to do.” One profound personal realization in my journey is recognizing that this desire is never fully satisfied in research. When you think you know, it's often because there's more to uncover.

Pedro E. Correia
Somewhere between Portugal and The Netherlands, December 2023

Executive Summary

The global increase in energy consumption and the demand for sustainable energy solutions have driven the growth of the offshore wind sector. As the number of installations continues to grow, the offshore-feeder concept has emerged as an efficient solution to streamline operations and reduce installation costs. This approach maintains installation vessels at sea while using cheaper feeder vessels for a continuous supply of components. Nevertheless, ship-to-ship (STS) cargo transfers are inherently hazardous and complex due to the unpredictable nature of the sea. Recognising this, maritime crane companies offer relative heave compensation (RHC) systems to assist crane operators in their operations. These systems mitigate the impact of heave motion between ships, known as relative heave, thereby enhancing safety and operational efficiency.

The existing state-of-practice in RHC systems involves using one Motion Reference Unit (MRU) on each vessel and a wireless communication link to measure relative heave. However, concerns have been raised by crane vessel owners regarding the dependence on sensors located on feeder vessels, citing issues such as the unavailability of MRU sensors on feeder vessels and the hurdles associated with sharing data among vessels owned by different companies. Moreover, when the required sensors are unavailable, the necessity of fitting temporary sensors onto supplier vessels further complicates STS transfer operations. Therefore, the research has been aimed at the following research question:

How can novel feeder-independent measuring methods be developed and assessed for relative heave estimation during offshore ship-to-ship cargo transfer operations?

This thesis proposes four innovative feeder-independent measuring solutions to address this research question. These solutions have been developed, implemented, and rigorously assessed for accuracy and processing time within a simulated environment.

Based on the existing literature, this study explores a novel sensor configuration consisting of an MRU and a 3D exteroceptive sensor placed on the crane boom tip for relative heave measurement. Two promising exteroceptive sensors, namely imaging Radar and LiDAR, emerged as viable options. Due to their shared potential and similarities, this research avoids making an exclusive choice and instead explores the applicability of both LiDAR-MRU and Radar-MRU sensor units.

Given the unavailability of actual sensor data, this study establishes a simulation workflow to generate synthetic sensor data. A co-simulation between Simulink and Unreal Engine is implemented to generate realistic sensor data in a dynamic 3D ship-to-ship environment. This data is stored alongside the relative heave distance and speed ground truth values derived from an exact relative heave kinematic model developed within this work for later evaluation.

This study introduces a range of relative heave measuring solutions, encompassing three key phases: Data Processing, Data Analysis, and Estimation. Three distinct measurement methods are developed: *deck detection*, *reflectors detection*, and *Doppler-based*, each with unique data processing and analysis methodologies. As a result, four distinct measurement solutions are implemented: *deck detection* (LiDAR-MRU), *deck detection* (Radar-MRU), *reflectors detection* (LiDAR-MRU), and *Doppler-based* (Radar-MRU). These solutions are supported by a Kalman Filter (KF) for relative heave distance and speed estimation.

The implemented solutions estimate relative heave distance with a Mean Absolute Error ranging from 0.3% to 5.2% of the reference's maximum amplitude and speed within the range of 0.8% to 4.3%. Notably, these solutions achieved relatively low computational times ranging from 1.9 to $91.7[ms]$, displaying potential to be integrated into real-time applications. The *reflectors detection* (LiDAR-MRU) method achieved the optimal balance between accuracy and processing speed, albeit reliant on reflective markers. The *deck detection* (LiDAR-MRU) method yielded the highest accuracy in distance estimates but obtained the slowest processing speed. On the other hand, the *Doppler-based* (Radar-MRU) method excelled in speed estimates but displayed susceptibility to distance estimates drift.

For future research, validating the simulation results using actual sensor data obtained in an up-to-scale setup is advised. Furthermore, fusing the developed methods is highly suggested to achieve more robust and accurate relative heave distance and speed estimates.

Contents

List of Figures

List of Tables

Nomenclature

Abbreviations

Abbreviation	Definition
3D	Three Dimensional
AHC	Active Heave Compensation
APT	Average Processing Time
CAD	Computer-Aided Design
DOF	Degrees of Freedom
DP	Dynamic Positioning
EKF	Extended Kalman Filter
FMCW	Frequency-Modulated Continuous Wave
FOV	Field Of View
IMU	Inertial Measurement Unit
IR	InfraRed
KF	Kalman Filter
KPI	Key Performance Indicators
LiDAR	Light Detection And Ranging
MAE	Mean Absolute Error
MLS	Moving Least Squares
MMW	MilliMeter Wave
MRU	Motion Reference Unit
NBF	Normal-based Bilateral Filter
PID	Proportional – Integral – Derivative
RADAR	RAdio Detection And Ranging
RANSAC	RANdom SAmple Consensus
RHC	Relative Heave Compensation
STS	Ship-to-Ship
TMC	Tub Mounted Crane
TOF	Time of Flight
UAV	Unmanned Aerial Vehicle
UE	Unreal Engine
USV	Unmanned Surface Vehicles
VG	Voxel Grid
WTIV	Wind Turbine Installation Vessel

1 Introduction

This chapter provides an introduction to the research project. It begins by presenting the research background in section 1.1, followed by an introduction to the research problem in section 1.2. Subsequently, section 1.3 defines the research goal, while section 1.4 discusses the research scope. Lastly, section 1.5 outlines the research questions, and section 1.6 offers the research outline.

1.1. Research Background

The ever-increasing global energy demand is expected to continue its upward trend in the coming years, as shown in Figure 1.1. This urgent demand, coupled with the pressing need to achieve ambitious energy transition goals, has accelerated the quest for sustainable energy sources. Offshore energy production has emerged as a promising alternative, evidenced by the growing number of wind farm installations in Figure 1.2. Such proliferation of wind farms has driven new installation projects further from the shore and into deeper waters [2]. Consequently, Wind Turbine Installation Vessels (WTIV) cover greater distances, increasing installation downtime and economic costs. To mitigate these challenges, improving efficiency and extending installation up-time is essential.

Figure 1.3 illustrates a promising offshore-feeder concept, where "*the installation vessel remains stationary, and components are delivered using cheaper transportation vessels*" [3]. Existing research [4, 5] has found potential installation cost reductions of up to 22% when employing the offshore-feeder approach compared to the conventional method, where the WTIV is solely responsible for transporting and installing all components. This finding underscores the importance of investigating and enhancing Ship-to-Ship (STS) cargo transfer operations for offshore-feeder applications. Furthermore, it is worth noting that the benefits of improving these operations extend beyond offshore wind turbine installations. For instance, pipeline installations where a continuous supply of pipes is required. Advancements in STS cargo transfer operations can improve these and various other offshore activities.

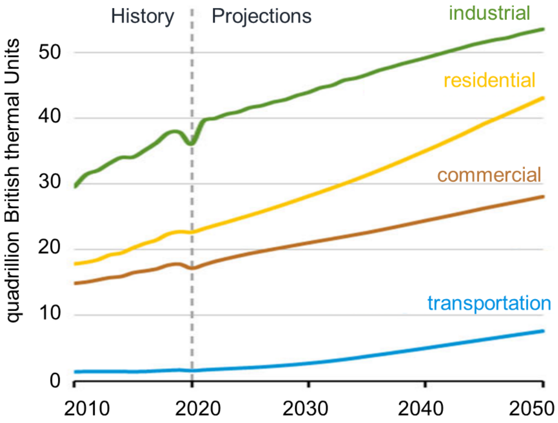


Figure 1.1: Worldwide electricity consumption end-use (2021). Figure retrieved from [6].

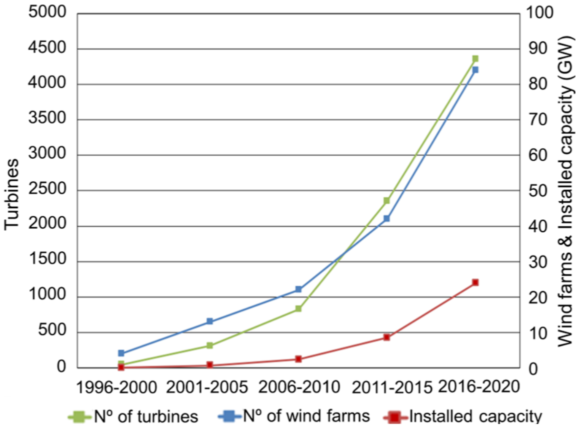


Figure 1.2: Evolution of offshore wind farms with respect to the commissioning period (2020). Figure retrieved from [7].

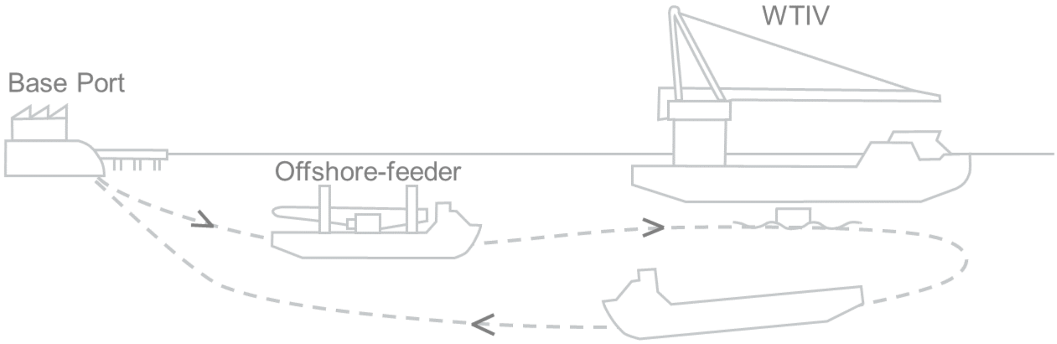
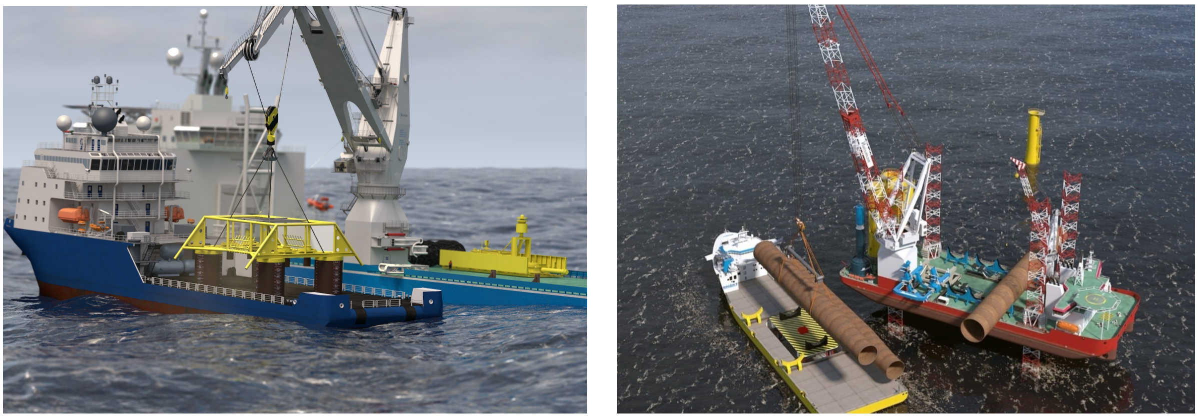


Figure 1.3: Offshore-feeder concept - The Wind Turbine Installation Vessel (WTIV) is stationed offshore, while smaller feeder vessels provide a constant supply of the required components for the installation.

Ship-to-Ship cargo transfer operations encompass the movement of various commodities, including crude oil, gas, bulk cargo, equipment, and installation components, between two seagoing vessels [8]. The approach to these transfers may vary depending on the nature of the cargo involved. However, this research specifically focuses on the transfer operations of components intended for offshore installations. In this scenario, cargo is transferred between ships using a ship-mounted crane, as illustrated in Figure 1.4. Therefore, throughout this thesis, when "STS cargo transfer" or "STS load transfer" terms are mentioned, it consistently refers to transferring components from one ship to another utilising a crane.



(a) STS transfer with Crane and Feeder vessels in motion

(b) STS transfer with moving Feeder vessel and fixed Jack-Up vessel

Figure 1.4: Ship-to-Ship (STS) cargo transfers in open sea. (a) STS transfer with both ships in motion. (b) Variation of (a) where one ship is fixed to the seabed and the other is moving. Figures retrieved from [1].

STS load transfer operations are pivotal to the offshore-feeder installation concept, obviating the need to load cargo at the nearest port and, thus, offering significant environmental and economic advantages. These benefits include reduced fuel consumption, exemption from port fees, shorter transfer times compared to traditional port operations, decreased reliance on specific port facilities, and circumvention of regulatory constraints, such as those stipulated by the Jones Act [9]. On the other hand, STS transfer operations are inherently complex and hazardous, requiring skilled crane operators to ensure the safety of cargo and personnel. These operations account for approximately 12% of offshore injuries [10] and are typically restricted to specific weather conditions, characterised by wave heights under 2.5 meters, often referred to as the "weather window" [11].

To alleviate the challenges associated with load-transferring tasks, automatic or autonomous systems can be implemented to assist crane operators. Incorporating advanced technologies, like Dynamic Positioning (DP) systems, load stabilisation and compensation systems, substantially improves STS cargo transfers [12]. They have the potential to expand the operational weather conditions, decrease reliance on limited specialised crane operators, and ultimately improve safety and consistency. This thesis explores one technology in particular named Relative Heave Compensation (RHC), an advanced technology derived from Active Heave Compensation (AHC) systems. RHC assists crane operators in cargo lift-off and touchdown procedures, reducing the risk of load collisions with the ship's deck and simplifying the load connection process. RHC systems address these challenges by actively compensating for the heave motion experienced by both ships, also known as relative heave.

The RHC system measures relative heave and adjusts the wire rope's length accordingly to maintain a constant distance between the hook and the supplier vessel's deck when the operator provides no other input. This system improves operators' precision during the handling operation, especially under rough sea conditions, by relieving operators from the need to account for the supplier vessel's continuous relative motion. For a visual representation of RHC, see Figure 1.5.

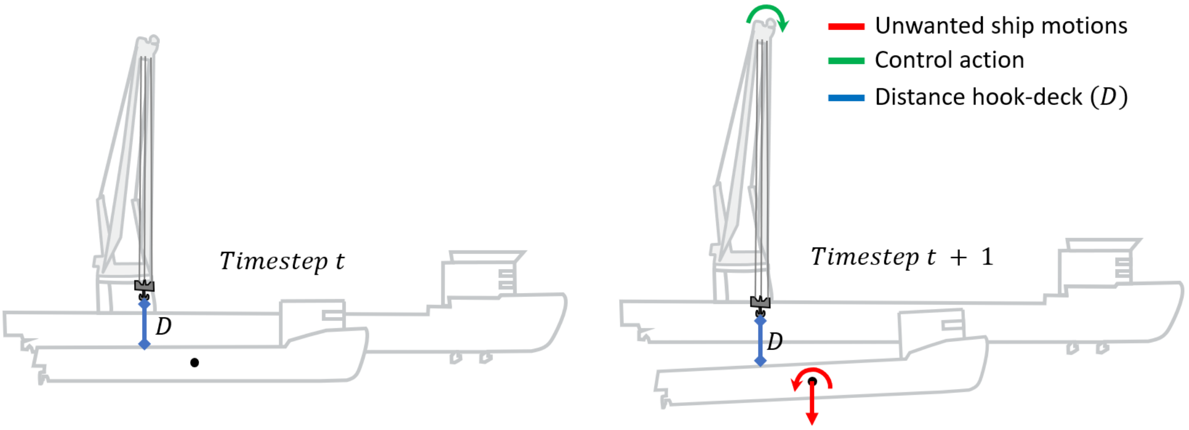


Figure 1.5: Schematic of Relative Heave Compensation (RHC) technology, which aims to keep the load/hook at a steady distance above the secondary ship deck.

RHC control schemes, as illustrated in Figure 1.6, consist of two essential components: the perception or measurement unit and the motor controller. The perception unit gathers crucial information about the system states, enabling the controller to regulate the paid-out wire for relative heave.

Despite several companies offering RHC solutions, including Huisman, MacGregor, and NOV [1, 13, 14], all use a similar measuring approach, where two Motion Reference Units (MRUs) are employed in combination with a wireless link to measure the heave motion of each ship and estimate relative heave. However, their reliance on the feeder vessels' sensors and communication protocols complicates the process for practical use. Notably, a feeder-independent solution has yet to be developed. Due to its significance and time constraints, this thesis exclusively focuses on investigating alternative relative heave measurement systems.

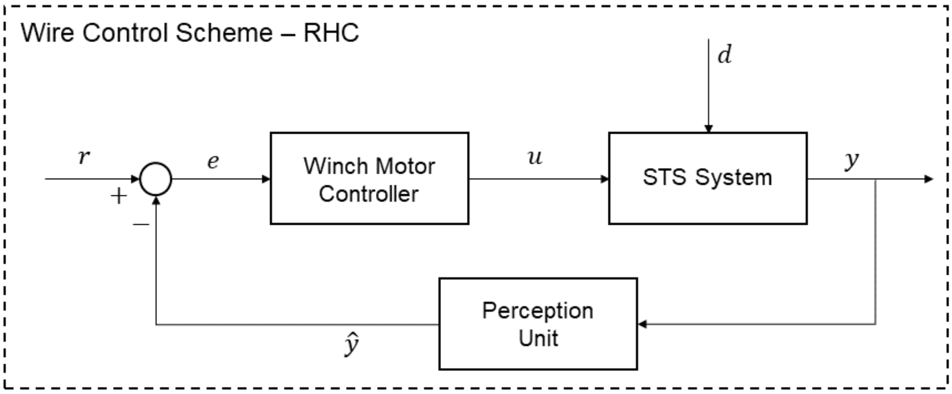


Figure 1.6: Schematic of a wire control system with an RHC assistant. Key variables: y - actual hook-to-deck distance, e - difference between set distance r and estimated distance $\hat{y}$, u - control input for the motor. The system is influenced by ship motion (d).

1.2. Problem Definition

One actor within the offshore industry looking to improve their RHC system is Huisman, where this research project has been performed. Huisman specialises in the design and manufacturing of ship-mounted cranes for a wide range of applications. The company is proactively committed to improving safety and operational efficiency through its innovative solutions.

As highlighted in the preceding section, Huisman, much like other industry leaders, currently offers the state-of-practice RHC system. This system currently measures relative heave employing two MRUs and a wireless communication link, as depicted in Figure 1.7.

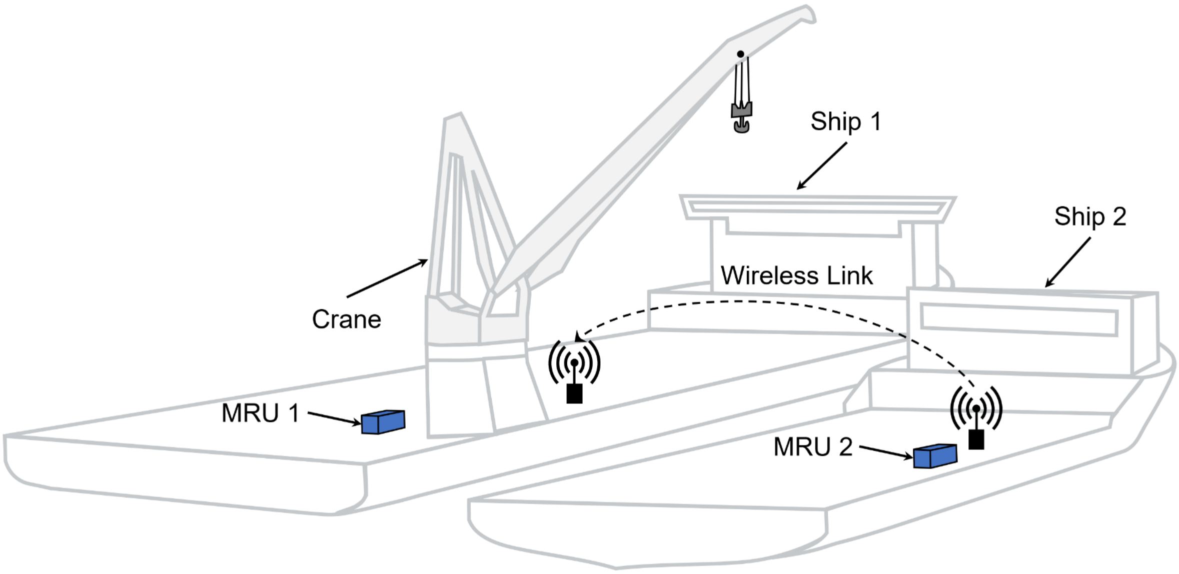


Figure 1.7: Relative Heave Compensation state of practice during Ship-to-Ship cargo transfer. The measurement system makes use of two MRUs and a wireless data link.

However, Huisman's clients have expressed concerns about deploying an RHC system that relies on sensors aboard feeder vessels. These reservations primarily arise from two key factors: the frequent absence of MRU sensors on feeder vessels and the bureaucratic complexities associated with sharing information among vessels owned by different companies. Moreover, when the required sensors are unavailable, the necessity of fitting temporary sensors onto supplier vessels further complicates STS transfer operations.

In light of these concerns, Huisman is proactively looking for alternative measurement solutions that are independent of the supplier's onboard sensors and can still be integrated into their current control system (Appendix B). Integration into Huisman's existing control scheme requires the measurement of two essential system states, herein referred to as: *relative heave distance* (D), utilised by the PID position controller, and its rate of change, denoted as *relative heave speed* ($\dot{D}$), employed as a feedforward gain. Refer to Figure 1.8 for a visual representation of these physical quantities.

In summary, the problem has been defined as follows:

Problem Statement

Currently, there are no established methods for measuring relative heave in the context of STS operations, solely relying on sensors installed on the crane vessel. Thus, these operations depend on the characteristics of feeder or supplier vessels, which often lack the required sensors. This reliance can add complexity and inefficiency to the operations, especially if temporary sensor placement is needed.

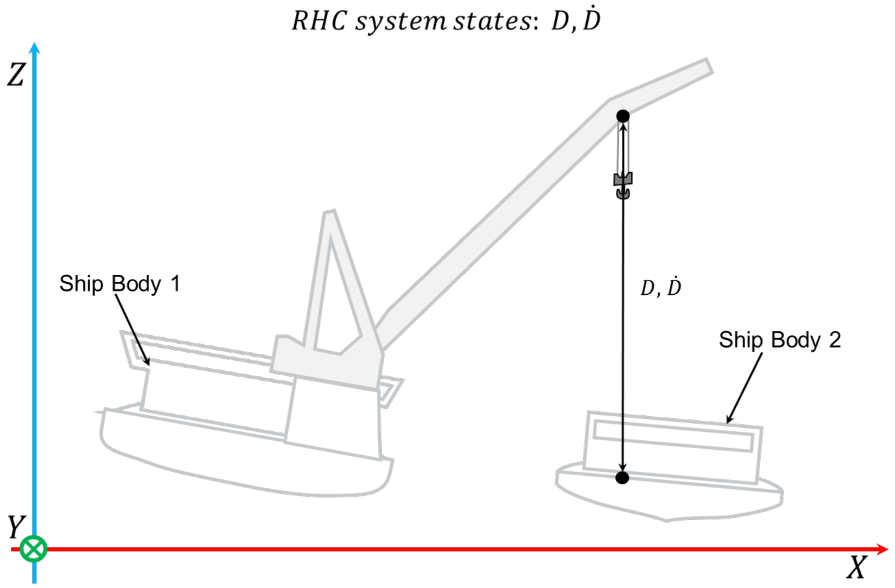


Figure 1.8: Illustration of the required system states for RHC, encompassing relative heave distance (D) and speed ($\dot{D}$), with D denoting the vertical separation between the boom tip and the supplier ship deck, and $\dot{D}$ signifying its rate of change.

1.3. Research Goal

Based on the defined problem, the goal of the research has been formulated as follows:

> **Research Goal**
>
> *This research aims to propose, implement and compare novel methods for measuring relative heave distance and speed in offshore Ship-to-Ship operations, relying exclusively on sensors installed on the crane vessel.*

In principle, to establish a novel feeder-independent solution, a different sensor unit must be proposed to replace the need for the external MRU placed on the supplier vessel. Once a different sensor unit is proposed, different measuring methods may be developed to process and analyse the sensor data to estimate relative heave distance and speed. The end goal will be implementing and comparing the developed methods to recommend a robust feeder-independent relative heave measuring solution.

1.4. Research Scope

In light of time constraints, a specific sensor configuration has been selected for the focus of this research. This chosen setup incorporates two key sensors: an MRU positioned on the crane vessel and an exteroceptive sensor affixed to the crane boom tip, as visualised in Figure 1.9. Following a thorough review of the available literature, two specific exteroceptive sensors, namely Light Detection and Ranging (LiDAR) and Radio Detection And Ranging (RADAR), were chosen for detailed investigation, resulting in the exploration of two distinct sensor units: the RADAR-MRU and the LiDAR-MRU.

Due to limitations in acquiring the proposed sensors, the development of measurement methods through experimental setup was not feasible. Therefore, a significant portion of this research is dedicated to creating a 3D simulation environment for generating the necessary sensor data to support the development of various measurement methods.

This study primarily focuses on developing measurement methods for relative heave estimation through sensor data processing and analysis. As a result, sensor modelling is not within the scope of this study, and pre-implemented sensor models from MATLAB were used to generate the necessary data. Additionally, the complex task of modelling ship and crane dynamics is beyond the scope of this work. Consequently, the 3D simulation environment assumes regular sinusoidal motion for each ship's six degrees of freedom and a fixed crane position.

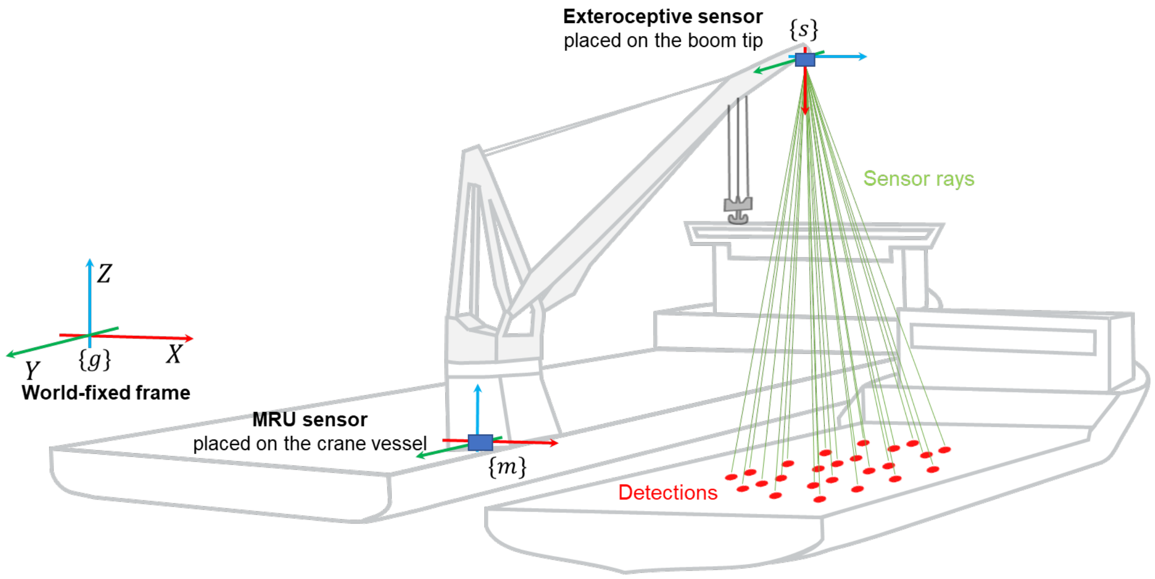


Figure 1.9: Configuration of the relative heave measuring system under investigation comprising two sensors: an MRU for measuring the crane vessel motion relative to a fixed reference and an exteroceptive sensor to measure the supplier vessel motion relative to the crane vessel.

This thesis introduces a kinematic model of the relative heave measurement problem, establishing a benchmark for assessing the implemented measurement solutions. Specifically, two methods have been employed for each sensor unit, yielding four solutions for comparative analysis regarding processing speed and accuracy.

1.5. Research Questions

In order to address the main research goal, the study has been structured according to a main research question along with several sub-questions:

Research questions

How can novel feeder-independent measuring methods be developed and assessed for relative heave estimation during offshore Ship-to-Ship cargo transfer operations?

1. What are the current limitations of relative heave measurement systems?
2. What sensor units provide a robust and supplier-independent alternative for relative heave measurement?
3. How can a kinematic model be formulated to represent relative heave during Ship-to-Ship operations, accounting for all six degrees of freedom of both vessels?
4. How can simulated sensor data be generated to support the development of a relative heave measuring system in the context of Ship-to-Ship operations?
5. How can measuring methods be developed to process and analyse the simulated data for real-time relative heave estimation?
6. How does the performance of the implemented measuring solutions compare against each other, considering accuracy and processing time?
7. What factors can influence the performance of the developed measurement methods?

1.6. Research Outline

Table 1.1 provides a clear overview of the report structure and highlights the chapters where each research question is addressed. Chapter 1 introduces the research area and outlines the main motivations behind investigating relative heave measurement systems. Chapter 2 delves into the state-of-the-art relative motion measurement systems, establishing the foundation for this work and supporting the selection of RADAR-MRU or LiDAR-MRU sensor units. These chapters address research questions 1 and 2. Chapter 3 focuses on developing a mathematical model to represent relative heave in STS operations. It also presents the simulation implementation of such scenarios for data collection using RADAR and LiDAR models, addressing research questions 3 and 4. Chapter 4 addresses research question 5 by presenting three distinct measuring methods and their corresponding implementations. Each implementation converts the simulated sensor data into relative heave distance and speed estimates using data processing and analysis algorithms. Chapter 5 offers an in-depth analysis of the performance of the different implementations and examines the influence of different parameters in the algorithms. This chapter comprehensively addresses both research questions 6 and 7. The final Chapter 6 concludes the work, summarising the main findings and providing recommendations for future research in this field.

Table 1.1: Report structure

Chapter	Content	Research Questions
1. Introduction	Motivation, Problem Definition	1
2. Literature Review	State-of-the-Art	2
3. Relative Heave Simulation and Modelling	Relative Heave Mathematical Model, STS Simulation Implementation	3, 4
4. Relative Heave Measurement and Estimation	Data Processing and Analysis, State Estimation	5
5. Measuring Solutions Verification and Evaluation	Methods Performance, Sensitivity Study	6, 7
6. Concluding Remarks	Addressed Research Questions, Recommendations	-

2 Literature Review

This chapter provides a comprehensive review of the latest applications for measuring relative heave in STS transfer operations. Its primary objective is to highlight advancements within both the industry and academia while also identifying existing research gaps. It begins with an overview of STS cargo transfer operations, encompassing industry practices, RHC systems, and relevant academic research. Subsequently, it delves into the measurement of relative motion in STS operations, exploring applicable sensors and the existing measurement methods, addressing **Research Question 2**. Lastly, the chapter presents pertinent data processing and analysis algorithms for the proposed sensor unit to address the challenge of measuring relative heave. The insights derived from this chapter will serve as the foundation for developing novel measuring methods.

2.1. Ship-to-Ship Cargo Transfer

STS cargo transfers involve the movement of various commodities, such as crude oil, gas, bulk cargo, equipment, and installation components, between two seagoing vessels [8]. The specific approach to these transfers may vary based on the nature of the cargo. Notably, it is crucial to acknowledge that these transfer operations are inherently complex and dangerous, requiring skilled crane operators to ensure the safety of both the cargo and personnel involved. Moreover, they are typically conducted under specific weather conditions, limited by wave heights of 2.5 meters [11].

Nevertheless, there has been a growing interest in STS load transfer operations due to their many advantages, including reduced fuel consumption, exemption from port fees, and the ability to circumvent regulatory constraints, such as those governed by the Jones Act [9]. This raised interest has led prominent companies like Huisman [1], MacGregor [13], and NOV [14] to create innovative products aimed at streamlining and enhancing Ship-to-Ship cargo transfer operations.

The industry solutions mainly aim at assisting crane operators, guaranteeing a smoother and safer transfer operation. One technology, in particular, that has gained significant attention is RHC, an advancement of active AHC systems. RHC is designed to aid crane operators during cargo lift-off and touchdown procedures, thereby reducing the risk of load collisions with the ship's deck and simplifying the load connection process. RHC systems address these challenges by actively compensating for the heave motion experienced by both vessels, commonly referred to as relative heave. The approach often involves utilising two MRUs with a wireless link to measure the relative heave motion, as depicted in Figure 1.7.

Tørdal's research [11] on Ship-to-Ship motion compensation highlights that while MRUs are capable of measuring all six Degrees Of Freedom (DOF), the compensation systems developed by MacGregor and NOV only consider the heave motion measured by the MRUs. This approach overlooks the significance of the roll and pitch movements experienced by the involved ships, which are also essential factors to consider when aiming to position the cargo on the second ship's deck accurately.

Taking a step further, Huisman [1] has proposed an advanced solution for Relative Heave Compensation. Their approach, presented in Appendix B, not only considers the heave motion of both ships but also incorporates the crane vessel's roll and pitch in their geometry calculation block. This enhanced approach provides a more accurate representation of the operation's dynamics. However, it does not account for the feeder ship's roll and pitch motions.

While the STS cargo transfer problem has gained considerable attention in the offshore industry, academic research in this area is still limited and not highly developed. Only McKenzie & Irani [15] and Tørdal [11] have contributed to the academic literature addressing the load-transferring problem directly, while an additional two other papers [16, 17] were discovered, which discuss Ship-to-Ship operations primarily focusing on the hydrodynamic interaction forces between ships rather than the load-transfer problem itself. Nevertheless, there are other valuable resources available, such as best practice guides [8, 18], along with regulations implemented by specific ports, including the port of Gothenburg [19], that offer insights into Ship-to-Ship transfers.

The presence of a mathematical model for the Ship-to-Ship transfer problem is valuable for gaining a comprehensive understanding of the problem, making exact system estimations, and devising effective control strategies for Ship-to-Ship compensation. Among the existing literature, Tørdal [11] stands out as the sole author who has developed such a model for STS operations, drawing from the previous ship modelling work done by Fossen [20] as a reference source. Although Tørdal's work does not explicitly address relative heave, it provides a valuable foundation for future research. Figure 2.1 depicts Tørdal's kinematic description of the STS cargo transfer problem. It is important to note that Tørdal heavily relies on MRUs placed on the secondary ship throughout his work.

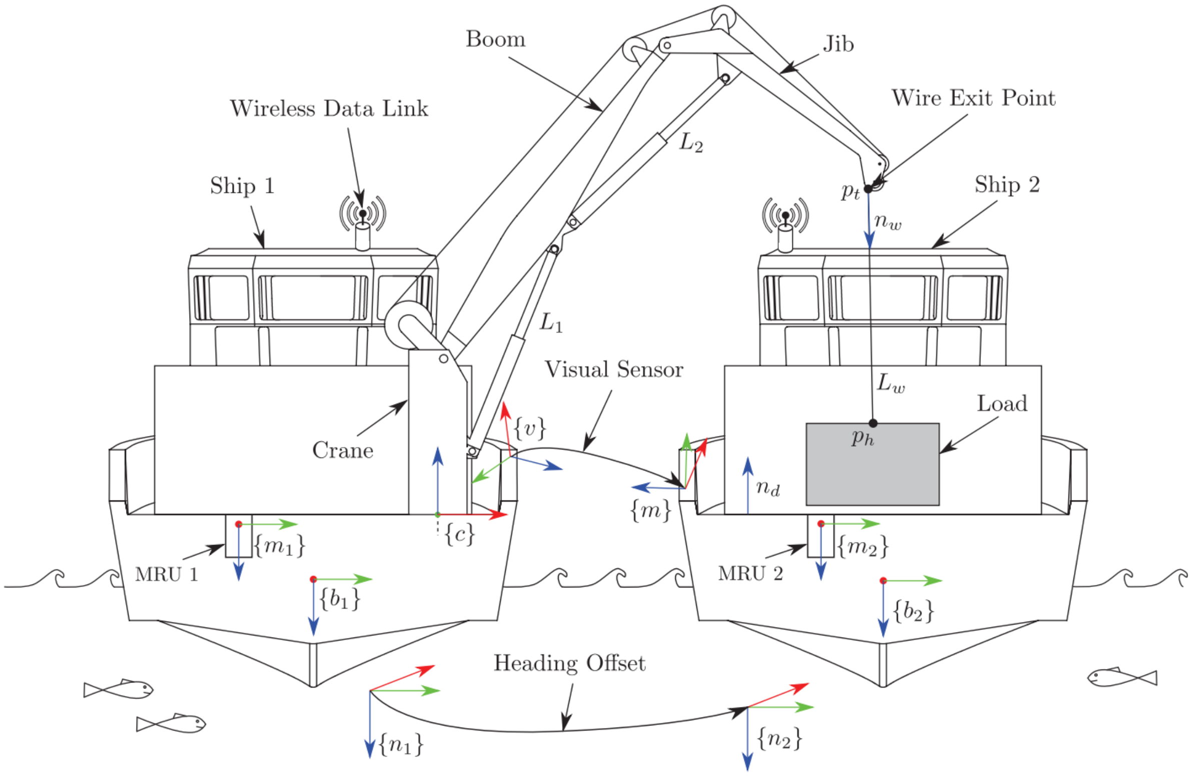


Figure 2.1: Illustration of the Ship-to-Ship motion tracking problem where two ships are laying alongside each other, and a suspended load is supposed to be landed on the secondary ship deck using an offshore crane. Figure retrieved from [11].

This section highlights the current state of STS transfer operations. Existing practical solutions do not comprehensively address the necessary ship motions in RHC, often relying on sensors placed on the supplier vessel. Bridging this gap is essential to enhance RHC systems. Despite available commercial solutions and general guidelines, academic research focused on improving STS operations is limited. Thus, there is a clear need for further research to model and develop RHC systems. The subsequent section 2.2 will delve deeper into measuring relative heave, exploring applicable sensors and existing measurement methods.

2.2. Relative Motion Measurement

Machine perception is essential for many automation applications, as it refers to how the machine or system perceives the surrounding environment. In the context of RHC systems, the crane vessel must accurately perceive the relative motion of the supplier vessel.

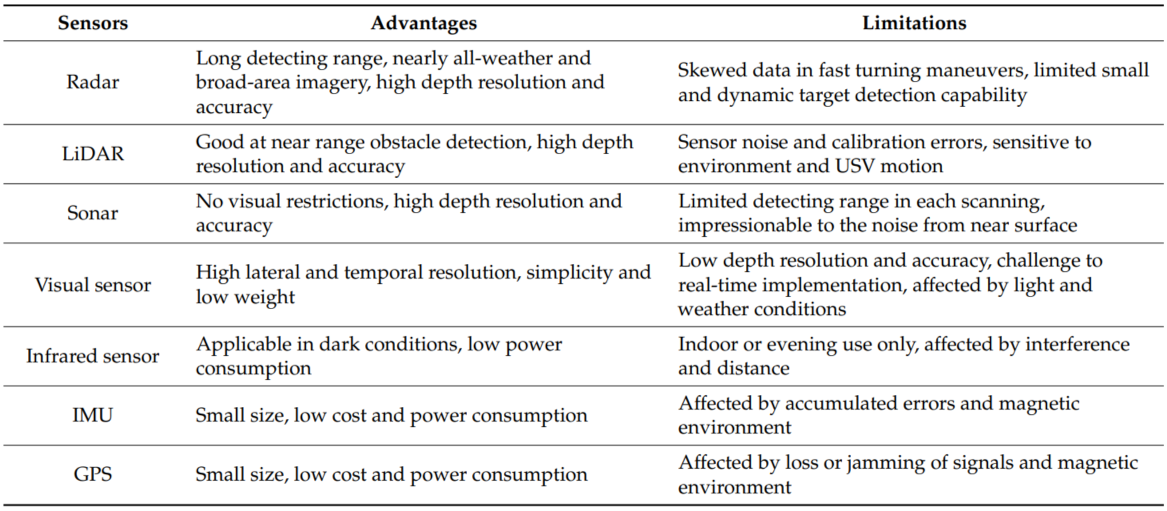
Multiple fields have researched relative motion or relative pose perception, including autonomous driving [21], robot grasping [22, 23], and unmanned surface vehicles (USVs) [24]. However, limited research applies it to offshore cranes.

In the forthcoming sections, subsection 2.2.1 discusses the pertinent sensors for relative motion perception, and subsection 2.2.2 presents and compares the various sensor combinations found in the literature.

2.2.1. Applicable Sensors

Various sensors are available, each with its own unique set of features and capabilities. An overview of these sensors and their potential applications can be found in the article by Balestrieri et al. [25], from which Table 2.1 was retrieved.

Table 2.1: Characteristics of USV sensors. Table retrieved from [25]



Sensors	Advantages	Limitations
Radar	Long detecting range, nearly all-weather and broad-area imagery, high depth resolution and accuracy	Skewed data in fast turning maneuvers, limited small and dynamic target detection capability
LiDAR	Good at near range obstacle detection, high depth resolution and accuracy	Sensor noise and calibration errors, sensitive to environment and USV motion
Sonar	No visual restrictions, high depth resolution and accuracy	Limited detecting range in each scanning, impressionable to the noise from near surface
Visual sensor	High lateral and temporal resolution, simplicity and low weight	Low depth resolution and accuracy, challenge to real-time implementation, affected by light and weather conditions
Infrared sensor	Applicable in dark conditions, low power consumption	Indoor or evening use only, affected by interference and distance
IMU	Small size, low cost and power consumption	Affected by accumulated errors and magnetic environment
GPS	Small size, low cost and power consumption	Affected by loss or jamming of signals and magnetic environment

Relative body motions refer to changes in a body's position, orientation, and acceleration over time. In order to measure these motions, sensors such as Inertial Measurement Units (IMUs) and Motion Reference Units can be attached directly to the body. Contrarily, exteroceptive sensors, including ultrasonic, infrared, LiDAR, RADAR, and stereo vision, can detect motion from a distance without being physically attached to the body.

The basic principles of operation, applications, advantages, and limitations of each sensor mentioned above will be discussed to provide a general understanding of the available sensors for measuring relative body motions.

IMUs and MRUs

IMUs and MRUs are motion sensors that measure the system's precise acceleration and angular velocity. These measurements are then used to derive the body's orientation, position and velocity.

IMUs are a combination of accelerometers, gyroscopes, and magnetometers [26]. Accelerometers measure acceleration in one, two, or three dimensions, while gyroscopes measure angular velocity in one, two, or three dimensions. Magnetometers, on the other hand, measure the strength and direction of a magnetic field, enabling the device to determine its heading. They are known for their high accuracy, low power consumption, and low cost. However, "IMUs suffer from serious errors in long-term position and orientation estimates due to the drift and the integration algorithm" [27].

MRUs, on the other hand, combine data from multiple sources, typically an IMU and a GPS receiver, to provide information about the absolute position and orientation of an object in space. Ultimately, this leads to lower drift and higher accuracy relative to IMUs. However, due to the higher complexity, MRUs are larger, more costly and consume more power. Furthermore, they rely on the availability of GPS signals. Even though the complexity arises, MRUs are still capable of real-time applications with a refresh rate of up to 200 Hz [28].

IMUs and MRUs are widely used in different applications, such as navigation [29], robotics [30], and control systems [31], as they provide reliable and accurate information about the object's position and orientation. However, it needs to integrate multiple motion sensors for applications where it is required to determine relative motions between two moving objects.

Ultrasonic

Ultrasonic sensors are relative motion sensors capable of measuring an object's distance and speed. They work by emitting a high-frequency sound wave and measuring the time it takes for the sound wave to bounce back after hitting an object. This time, known as the Time Of Flight (TOF), is used to calculate the distance to the object [32].

One of the main advantages of ultrasonic sensors is their high-precision velocity measurement. Fast sensor response is also a plus [33]. However, they have a limited measuring range of up to 11 m [34]. These sensors are sensitive to the environment, particularly to temperature and humidity variations. These conditions can affect the accuracy of the measurements. Additionally, ultrasonic sensors may be unable to measure through solid objects or materials with high absorption, such as water.

Ultrasonic sensors are commonly used in applications where the environment can be well-controlled, such as industrial automation [33], robotics [32], and automotive [35]. They are also known for low power consumption and low cost.

Infrared

Infrared (IR) sensors are relative motion sensors that use infrared radiation to detect the presence and motion of objects. They work by emitting or detecting infrared radiation and measuring the amount of radiation reflected or absorbed by an object. This information determines the object's distance and speed.

IR sensors can be classified into two main types: active and passive. Active IR sensors emit infrared radiation and measure the amount of radiation reflected by an object. They are employed in distance measurement, object detection, and security systems. On the other hand, passive IR sensors detect infrared radiation emitted by an object. One of the main advantages of IR sensors is their ability to detect objects in low light conditions or complete darkness. However, they have a relatively short range, similar to ultrasonic. According to Zappi et al. [36], it can detect up to 8 meters. IR sensors are also sensitive to environmental factors such as temperature, humidity, and background radiation, which can affect the accuracy of the measurements. Additionally, IR sensors may be unable to detect objects that do not emit or reflect infrared radiation, such as certain types of plastic or glass.

IR sensors are employed in a variety of applications such as robotics [37], automotive [38], and security systems [36]. They are also known for their high accuracy, low power consumption, and low cost.

LiDAR

LiDARs are relative motion sensors that use laser light to measure the distance and shape of objects. They work by emitting a laser beam and measuring the time it takes for the beam to bounce back after hitting an object. The TOF is used to calculate the distance to the object. Additionally, by analysing the pattern of the reflected laser beams, LiDAR sensors can also determine object shape and surface characteristics.

LiDAR sensors classify into two main types: mechanical LiDARs and solid-state LiDARs. The former uses a rotating assembly to create a wide Field of View (FOV), while the latter has no spinning components, increasing the sensor robustness but reducing the FOV [39]. Solid-state LiDARs include a variety of different methodologies. However, it is crucial to highlight the Frequency-Modulated Continuous Wave (FMCW) LiDAR. This type of LiDAR does not use the typical TOF principle. It emits a continuous wave, which enables the system to measure distances and velocities to targets based on the Doppler principle [39].

One of the main advantages of LiDAR sensors is their spatial resolution. They are capable of capturing detailed 3D scenes of objects and their environment. LiDARs produce millions of points in a single scan, increasing robustness in case some points are faulty. Additionally, they have a wide FOV and relatively long-range measuring distances of up to several hundred meters [40]. On the other hand, LiDAR sensors are sensitive to environmental factors such as sunlight, fog, and dust, which can affect the accuracy of the measurements, yet less sensitive than some of the previously mentioned sensors. Their robustness will depend on the beam wavelength used. Furthermore, LiDAR sensors may not be able to detect objects that are transparent or have low reflectivity.

A variety of applications use LiDAR sensors, for instance, autonomous vehicles [40, 41, 42] and spacecraft [43, 44].

Millimeter-Wave (MMW) RADAR

RADARs, or radio detection and ranging systems, use radio waves to detect and locate objects. They work by emitting a radio frequency signal and measuring the time it takes to bounce back after it hits an object. After a series of signal processing techniques, "the RADAR can not only obtain the exact distance of multiple targets but also measure the relative velocity by the Doppler shift effect" [45].

RADARs have many advantages when it comes to measuring relative position and velocity. Compared to other sensors, "MMW-RADAR has a longer wavelength, certain anti-blocking, and anti-pollution ability, which can cope with rain, snow, fog, and dark environment" [45]. They can also detect objects at a long range, making them useful for monitoring large areas. Additionally, RADARs can track relative velocities without needing complex post-processing algorithms. However, RADARs also have some disadvantages. One of the main disadvantages is that they can be affected by interference from other radio waves. Additionally, RADAR's resolution can be challenging to distinguish between different close objects, making it hard to track multiple targets.

RADARs have a range of applications, including aviation [46] and autonomous vehicles [47, 48]. In the automotive industry, they can detect and track other cars or obstacles, which is essential for driver assistance systems [49].

Stereo Vision

Stereo vision sensors use two or more cameras to capture images of an object from different viewpoints. These images are then used to calculate the depth and 3D position of the object.

Stereo vision sensors work by analysing the differences in the images captured by the cameras, known as stereo correspondences, to calculate the depth of the objects in the scene. The stereo correspondences are typically found by matching image features, such as edges or corners. Once the correspondences are found, triangulation algorithms are used to calculate the depth of the objects. One of the main advantages of stereo vision sensors is their ability to provide a detailed 3D representation of an object and its environment. Additionally, they are relatively low-cost and can be easily integrated into a wide range of systems. However, stereo vision sensors can be affected by lighting conditions, camera calibration [50], and the presence of textureless or reflective surfaces. They also tend to be computationally expensive and have low accuracy, caused by poor depth estimation and limited range [51].

Stereo vision sensors are commonly used in applications such as robotics [52] and autonomous vehicles [53]. They can be integrated with other sensors, such as LiDAR or RADAR, to improve their performance and provide a more comprehensive representation of the object's position and orientation.

Sensors Offshore Applicability

From the sensors presented, it is relevant to analyse which sensors are suitable for relative motion estimation of a feeder vessel relative to the crane vessel in an offshore environment. This analysis was based on sensors available on the market and presented on Table 2.2.

The offshore industry is characterised by its use of large machinery, which requires sensors that can measure up to 200m distances between the crane's hook and the target position [1]. From Table 2.2, the use of stereo vision, infrared, and ultrasonic sensors in these applications is limited due to their ranging capabilities. Additionally, the offshore industry's harsh environment, vessel motions, and vibrations pose challenges for sensor solutions that require extensive calibration. This issue is particularly true for mechanical rotating LiDARs, which are less robust than their solid-state counterparts [39].

Table 2.2: Applicable sensors available in the market and their characteristics

	Common and/or Sensor Example	Range	Resolution	FOV	Applications	Benefits and Obstacles for Relative Motions Sensing
IMU/MRU	Inertial Labs - MRU-P [54]	coop. dep.	5 cm,.03 ms^{-1},.005 ms^{-2}	-	Ship navigation, Dynamic Positioning, Heave Compensation, Autopilot systems.	**Advantages:** (1) highly accurate; (2) highly robust to adverse conditions; (3) position, velocity, acceleration measurements. **Disadvantages:** (1) requires integration with other sensors to retrieve relative motions; (2) agent-cooperation dependent; (3) susceptible to drift.
	Kongsberg - MRU H [28]	coop. dep.	5 cm,.01 ms^{-1},.01 ms^{-2}	-		
	SMC - IMU-10 [55]	coop. dep.	5 cm, .01 ms^{-2}	-		
Ultrasonic	Baumer - UNAM 70U6131/S14 [56]	6 m	0.3cm	-	Level measurement, Distance measurement, and Object detection.	**Advantages:** (1) highly accurate; (2) robust to dust and dirt ; (3) velocity can be extracted from doppler effect. **Disadvantages:** (1) limited range; (2) Interference from other ultrasonic sources; (3) Sensitivity to humidity and temperature.
	Omron - E4PA-LS600 [57]	6 m	0.1 cm	-		
Infrared	Sharp - GP2Y0A710K0F [58]	5.5 m	-	-	Motion detection, parking assistance, distance measurement, and obstacle detection.	**Advantages:** (1) accurate; (2) robust to dust and dirt . **Disadvantages:** (1) limited range; (2) not suitable for velocity measuring; (3) require a highly reflective surface.
	Sharp - GP2Y0D02YK0F [59]	0.8 m	-	-		
LR LiDAR	Aeva - Aeries II (doppler) [60]	<500 m	-	120°x30°	Navigation, tracking and classification purposes in automotive, aerospace, trucking and heavy machinery environments.	**Advantages:** (1) high range and environment coverage; (2) 3D mapping and intensity information; (3) highly accurate; (4) imune to lighting conditions. **Disadvantages:** (1) high cost; (2) weather dependent (fog, rain, snoe); (3) require complex point cloud processing. Notes: (1) Doppler LiDARs can directly measure velocity; (2) high reflective targets can be used for object tracking.
	Baraja - Spectrum HD25 (doppler) [61]	250 m	0.04°x0.01258	120°x25°		
	Cepton - Vista-X120 Plus [62]	200 m	0.1°x0.1°	120°x25°		
	Hesai - AT128 [63]	200 m	0.1°x0.2°	120°x25.4°		
	Innoviz - InnovizTwo [64]	300 m	0.05°x0.05°	120°x40°		
	Luminar Iris [65]	250 m	0.05°x0.05°	120°x28°		
	Ouster - OS2 (M) [66]	200 m	0.18°x0.18°	360°x22.5°		
	RoboSense - RS-LiDAR M1 [67]	200m	0.2°x0.2°	120°x25°		
	Velodyne - Alpha Prime (M) [68]	300m	0.11°x0.4°	360° x 40°		
	Zvision - ML-Xs [69]	250m	0.13°x0.2°	120°x25°		
RADAR	Bosch - Front RADAR sensor (4D) [70]	302 m	0.1°	60°x12°	Collision detection, pedestrian detection, adaptive cruise control, speed monitoring.	**Advantages:** (1) high range; (2) robust to weather conditions; (3) accurate; (4) direct velocity measurement. **Disadvantages:** (1) limited FOV; (2) low resolution;
	Continental - ARS540 (4D) [71]	300 m	0.1°	60°x - °		
	Continental - ARS441 [72]	250 m	-	9°		
Stero Vision	Intel RealSense Depth Camera D435 [73]	3 m	2MP	69°x42°	Localisation and mapping, Object detection, classification and tracking.	**Advantages:** (1) high resolution images ideal for detecting and tracking objects; (2) Color and texture image data; (3) 3D mapping. **Disadvantages:** (1) limited range; (2) Accuracy decades significantly with depth; (3) High sensitivity to lighting and weather conditions; (4) Depht estimation is computional expensive process.
	Orbbec - Astra Pro [74]	8 m	-	60°x49.5°		
	StereoLabs - ZED 2 [75]	20 m	4MP	110°x70°		

Note: (-) not specified/applicable; (coop. dep.) cooperation dependent; (LR LiDAR) Long Range LiDAR; (M) Mechanical LiDAR; All the sensors are off-the-shelf products; an in-house developed Stereo Vision solution can achieve higher ranges; The sensors' refresh rate is not presented as every sensor has adequate refresh capacity for the problem being discussed.

For highly accurate position measurements and complete environment perception, LiDAR seems to be the most applicable, according to the information in Table 2.2. On the other hand, for fast and precise relative velocity measurements, RADAR is a sensible option due to its capacity for instantaneous speed measurements. MRUs combine accurate velocity and position measurements of the body where it is placed, but their application is not trivial for vessels' relative motion measurements.

According to Huisman [1], position and velocity are valuable inputs for Relative Heave Compensation control algorithms. From the previous statements, there is no clear best sensor, which suggests that a solution comprising multiple sensors will most likely achieve the best results. Furthermore, sensor solutions must function under harsh weather conditions to operate in the offshore industry. Combining sensors like MRUs and RADARs that suffer negligible effects from rain, fog, and snow with sensors like LiDARs capable of providing a comprehensive understanding of the environment is crucial. An example is found in the autonomous automotive industry which faces similar issues and has proven that optimal performance requires sensor fusion [21].

In summary, the offshore industry requires sensors with high-ranging capabilities and robustness to handle harsh weather conditions and vessel motions. LiDAR, RADAR, and MRUs are some of the most applicable sensors for this industry. However, their respective limitations suggest that a solution comprising multiple sensors will likely achieve the best results. The following subsection 2.2.2 will provide existing sensor solutions and measuring methods for Ship-to-Ship relative motion compensation.

2.2.2. Measuring Methods

Accurately measuring relative motions between ships is vital for advancing STS transfer technology. However, limited research is available on measuring relative motions in such scenarios. Among the authors who have studied STS load transfer, McKenzie & Irani [15] emphasise the need for a robust sensor solution to track the relative motions of the second ship. Notably, only Tørdal [11] has dedicated substantial efforts to address this specific issue, publishing a series of papers [76, 77, 78, 79] that analyse various sensors and configurations, demonstrating some success in measuring these relative motions. Although this thesis primarily focuses on measuring the relative heave motion rather than all relative motions, it is necessary to acknowledge that Tørdal's existing work will provide valuable inspiration for exploring measurement concepts within this research.

Tørdal's [76] initial solution, depicted in Figure 2.2, utilised two MRUs (one on each ship) and a wireless data link to measure relative motions, which has become the industry standard. This pioneering research successfully demonstrated the potential of the wireless data link to compensate for relative movements. However, challenges emerged regarding the measurement of absolute offset position and pose between the vessels, which the author identified as a crucial practical issue.

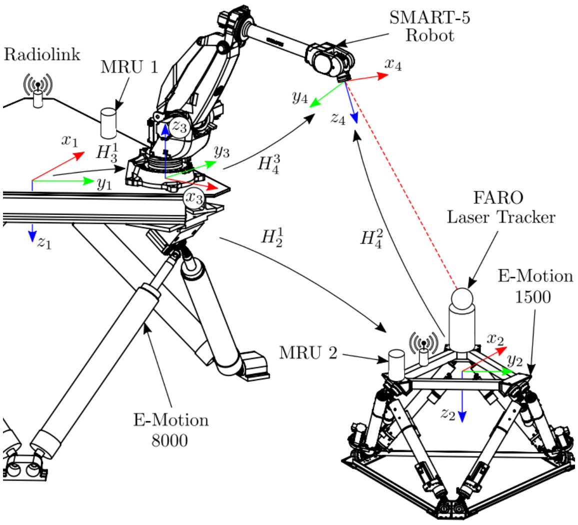


Figure 2.2: Experimental setup using two MRUs. Figure retrieved from [76]
Note: FARO laser tracker only purpose is to measure the absolute position error.

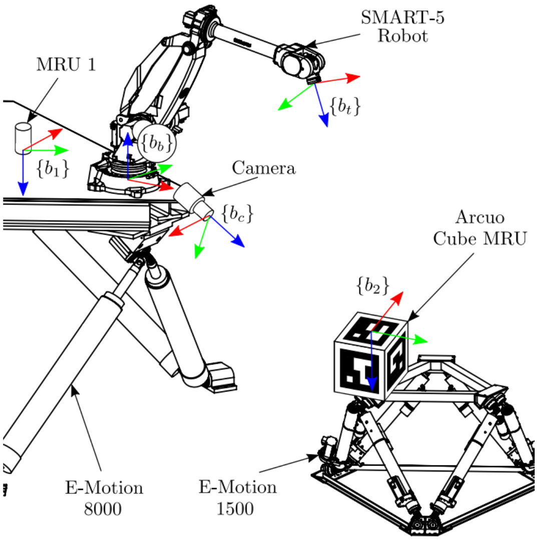


Figure 2.3: Experimental setup using two MRUs, a camera and an Aruco Cube marker. Figure retrieved from [77]
Note: An MRU is placed inside the Aruco Cube.

To address this concern, Tørdal proposed integrating a third sensor like vision, 3D, acoustic, or laser sensors to ensure precise measurement of the vessel's absolute offset position and attitude. In his subsequent publication, Tørdal expanded his initial work by incorporating a third camera sensor [77], as illustrated in Figure 2.3. This additional sensor enabled the measurement of the absolute distance between the two platforms used in the experimental setup. Nonetheless, the researcher acknowledged the importance of addressing the challenges related to camera processing time latency and camera calibration.

In light of previous findings, Tørdal [78] proposed a solution utilising a single time-of-flight sensor, specifically, a laser tracker, to accurately measure the six DOF of an external marker attached to the secondary platform or ship (refer to Figure 2.4). This research demonstrated the effectiveness of time-of-flight sensors in addressing the problem. However, it is essential to note that the sensor used in the study is primarily intended for indoor applications with a clear line of sight between the tracker and the marker on the secondary vessel. Therefore, the Tørdal suggests exploring more robust sensor technologies that may offer slightly lower accuracy but are still practical for implementation. Subsequent research [79] combined the laser tracker sensor with two MRUs (refer to Figure 2.5) using an Extended Kalman Filter (EKF) to track relative motions, even when laser tracker measurements were unavailable. Despite enhancing the system's robustness, finding a suitable time-of-flight sensor to measure these motions remained unresolved.

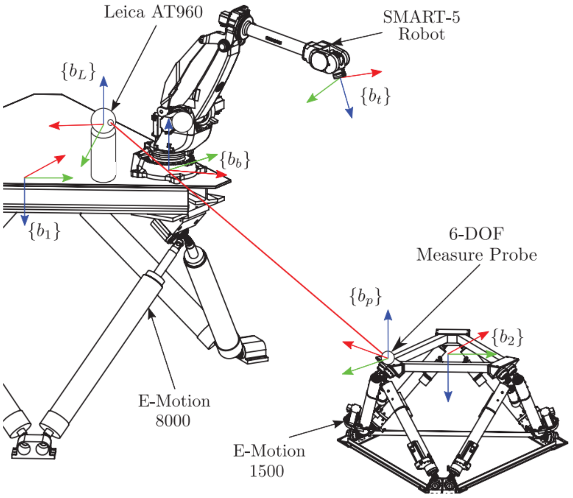


Figure 2.4: Experimental setup using a Laser Tracker. Figure retrieved from [78]

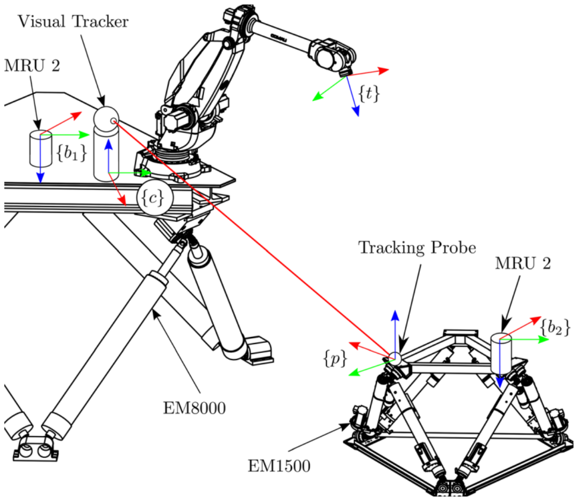


Figure 2.5: Experimental setup using two MRUs and a Laser Tracker. Figure retrieved from [79]

The reviewed literature primarily relies on external sensors mounted on the supply vessel to estimate relative motion, which poses practical challenges and limitations. However, Holen, Sjøberg, and Egeland [80] propose a system that overcomes this by combining a camera and a LiDAR for ship deck estimation without the need for external sensors, as depicted in Figure 2.6. Their approach, utilising line detection and a vanishing point detector, offers robustness and resistance to occlusions. The camera detects the ship deck while a single line of sight LiDAR measures the absolute distance. The obtained experimental results proved the system's potential for real-world implementation. Nonetheless, using a camera as the primary sensor for offshore industrial applications still poses significant challenges, including limited depth resolution, high computational resources, and vulnerability to adverse weather conditions. Further research is necessary to identify a more appropriate sensor for such applications.

Table 2.3 provides an overview of all the available measuring solutions for relative motion compensation, outlining the sensors utilised in each and summarising their respective advantages and disadvantages. Upon examining Table 2.2 and 2.3, it is further confirmed the notion that no individual sensor is ideal, and the optimal strategy involves combining several sensors. In a comparable scenario, Wang and Zhang [81] arrived at the same conclusion, noted in Table 2.4. The authors determined that for precise ship positioning and orientation estimation of berthing parameters, a LiDAR-RADAR fusion approach is most appropriate based on pertinent criteria.

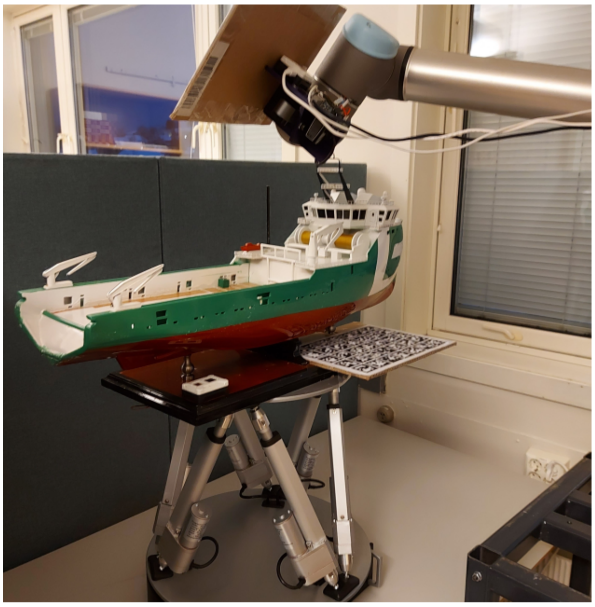


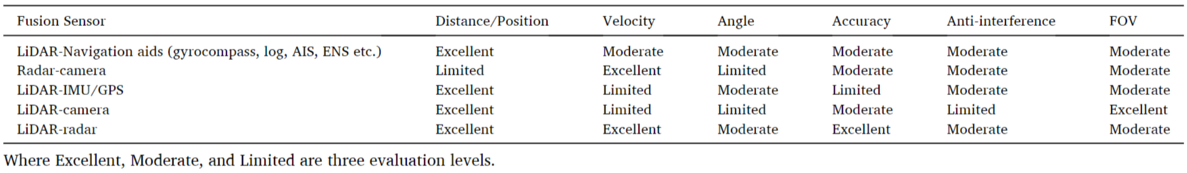
Figure 2.6: Experimental setup using a camera and single line of sight LiDAR. Figure retrieved from [80]

Table 2.3: Comparison of the various measuring solutions for relative motions between two vessels found in the literature

Ref.	Crane Vessel				Supply Vessel		Advantages	Disadvantages
	MRU/IMU	Camera	Laser Tracker	LiDAR	MRU	Marker		
[76, 82, 83]	✓	✗	✗	✗	✓	✗	6DOF pose, low complexity	communication link, unknown absolute distance between vessels
[77, 84]	✓	✓	✗	✗	✓	✓	6DOF pose, absolute distance between vessels	communication link, camera calibration, marker dependent
[78]	✗	✗	✓	✗	✗	✓	6DOF pose, absolute distance between vessels, high accuracy, communication independent	marker dependent , low robustness
[79]	✓	✗	✓	✗	✓	✓	6DOF pose, absolute distance between vessels, high accuracy	marker dependent
[80]	✓	✓	✗	✓	✗	✓	6DOF pose, absolute distance between vessels, high accuracy	marker dependent

Note: (✓) used; (✗) not used

Table 2.4: Various fusion sensors technology in calculating berthing parameters. Table retrieved from [81].



Fusion Sensor	Distance/Position	Velocity	Angle	Accuracy	Anti-interference	FOV
LiDAR-Navigation aids (gyrocompass, log, AIS, ENS etc.)	Excellent	Moderate	Moderate	Moderate	Moderate	Moderate
Radar-camera	Limited	Excellent	Limited	Moderate	Moderate	Moderate
LiDAR-IMU/GPS	Excellent	Limited	Moderate	Limited	Moderate	Moderate
LiDAR-camera	Excellent	Limited	Limited	Moderate	Limited	Excellent
LiDAR-radar	Excellent	Excellent	Moderate	Excellent	Moderate	Moderate

Where Excellent, Moderate, and Limited are three evaluation levels.

This subsection provides a more in-depth analysis of the existing measuring solutions found in the literature. According to the overview of the solutions presented in Table 2.3, it is apparent that current solutions for estimating relative motion heavily rely on external sensors or fixed markers installed on the supply vessel or platform. These approaches often require communication devices and protocols between the crane vessel and the supplier's vessel system, which can pose practical challenges during implementation. Furthermore, secondary vessels often lack the required onboard sensors, and the bureaucratic obstacles associated with information-sharing protocols for ships owned by different companies create additional difficulties. While Tørdal's research [11] suggested that vision or 3D sensors could overcome these challenges, only Holen, Sjøberg, and Egeland [80] have explored the use of vision as a base sensor for a supplier-independent solution. However, this is not an ideal approach due to the inherent limitations of the sensor. Therefore, there is a clear research gap in investigating measuring solutions independent of the supplier vessel, which employ 3D sensors capable of detecting the ship deck and accurately estimating its real-time position. From the 3D sensors discussed in this section, the author believes a sensor solution comprising LiDAR-MRU, RADAR-MRU or even LiDAR-RADAR-MRU should be investigated. As the sensors do not measure relative heave distance and speed directly, the following section 2.3 will discuss applicable data processing and analysis algorithms relevant to compute relative heave.

2.3. Data Processing and Analysis

For Relative Heave Compensation, accurately measuring the movements of supplier vessels is crucial. Among the suitable sensor options discussed in section 2.2 for this purpose, LiDAR and RADAR stand out as the most applicable choices. These sensors produce output in the form of a point cloud, which is essentially a collection of data points situated in 3D space. Each data point represents specific spatial coordinates (X, Y, Z) in the given space and may incorporate additional information, such as reflectivity or Doppler velocity measurements.

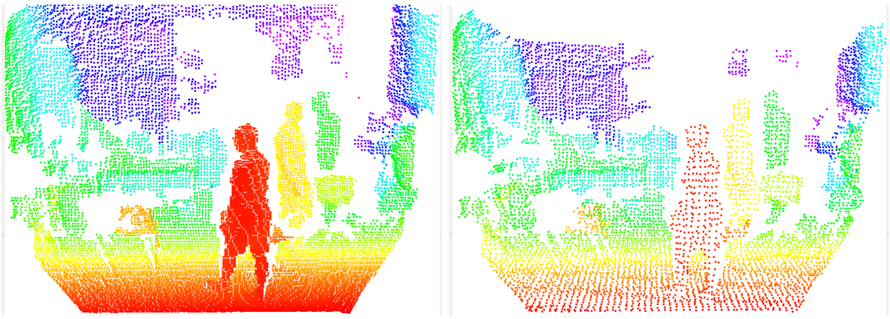
However, extracting relative motions from point clouds is a complex task, requiring tailored data processing and analysis techniques. This chapter offers an overview of appropriate algorithms to extract relative positions and velocities from point cloud sensor data. Section 2.3.1 provides an insight into various point cloud filtering techniques, while Section 2.3.2 reviews essential point cloud segmentation algorithms, both of which are pivotal for tracking vessel and deck movements.

2.3.1. Point Cloud Filtering

Point cloud data acquired from different sensors can be noisy and incomplete due to various factors, such as sensor noise, occlusions, and weather conditions. Therefore, point cloud filtering techniques are commonly used to remove noise and outliers from the raw point cloud data. According to a recent review [85], point cloud filtering techniques are essential for the performance of subsequent processing steps and time complexity reduction. This section describes four state-of-the-art and widely cited filtering algorithms suggested and analysed by Han et al. [85].

Voxel Grid (VG)

Voxel Grid (VG) filtering divides a 3D space into small, equally-sized cubes (voxels) and approximates points within each voxel by a single representative point. This preprocessing technique helps reduce the point cloud's complexity and can save computational cost in subsequent processing steps. However, VG filtering may cause geometric information loss, so voxel size must be chosen carefully to balance information loss and computation time.



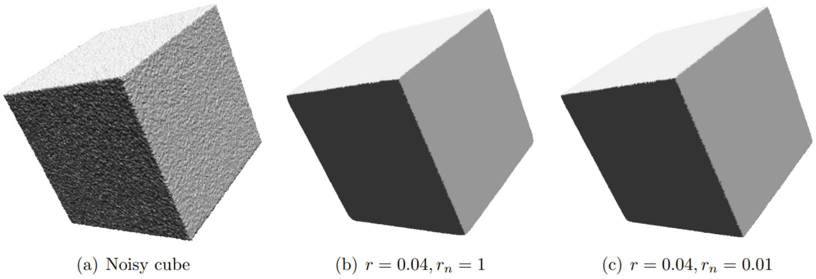
(a) Raw Kinect point cloud.

(b) After voxel grid filtering.

Figure 2.7: The effect of the voxel grid filter. Figure retrieved from [86].

Normal-based Bilateral Filter (NBF)

Normal-based Bilateral Filter (NBF) algorithm is a point cloud filtering technique that uses point surface normals to smooth the surface while preserving sharp edges. Digne and Franchis [87] proposed an efficient NBF algorithm that defines a weight function based on the distance and normal difference between neighbouring points and computes a weighted average of the points within a local neighbourhood using the weight function. NBF reduces noise in the point cloud while preserving its structural features (Figure 2.8). However, choosing appropriate parameters for the weight function can be challenging.



(a) Noisy cube (b) $r = 0.04, r_n = 1$ (c) $r = 0.04, r_n = 0.01$

Figure 2.8: The effect of the NBF on (a) a noisy cube (with variance equal to 0.1% of the box size) with one iteration with various parameters (b) and (c). Figure retrieved from [87].

Moving Least Squares (MLS)

Moving Least Squares is a point cloud filtering technique, initially proposed by Alexa et al. [88], that uses local regression to approximate the surface of the point cloud. MLS fits a polynomial function to a small set of neighbouring points around each point in the point cloud and approximates the surface by taking the weighted average of these local polynomial approximations. Figure 2.9 provides a graphical representation of the described algorithm. MLS is a low-pass filter that can smooth the surface of the point cloud and reduce noise but may also smooth out shape features, so it requires careful parameter tuning for optimal performance.

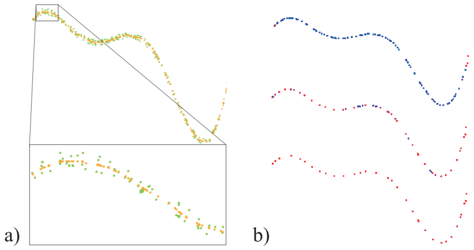


Figure 2.9: Figure (a) shows a close-up view of the MLS algorithm result (orange points) on a noisy input (green points). Figure (b) shows the iterative filtering process that eliminates points that do not contribute to the shape (blue points), keeping the important ones (red points). Figure retrieved from [88].

L0 minimisation

L0 minimisation is a denoising technique for point clouds that aims to preserve sharp features while reducing noise (Figure 2.10). It estimates normals using an L0 optimisation procedure and repositions points along the estimated normal directions to maintain sharp features. Sun et al. [89] presents an L0 algorithm that can achieve better results than the previous denoising techniques for point clouds with sharp features but is more computationally intensive and requires careful parameter tuning.

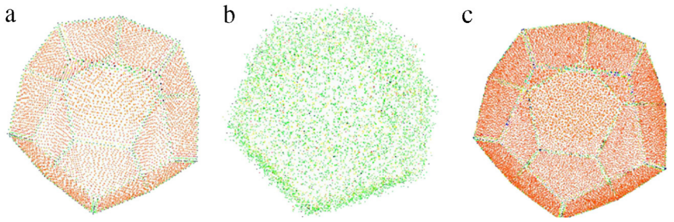


Figure 2.10: Denoise point sets via L0 minimisation. (a) Original; (b) input; (c) L0 minimisation. Figure adapted from [89].

2.3.2. Point Cloud Segmentation

Point cloud segmentation is the process of grouping points into homogeneous regions with similar properties or clusters that likely belong to the same object. After reviewing the most popular segmentation methods, Grilli et al. [90] distinguish segmentation methods into five main categories, shown in Figure 2.11.

Based on their review, point cloud segmentation can be done using edge-based and region-growing methods without prior knowledge of the environment. However, region-growing is more noise-resistant, leading to a more precise segmentation. Model fitting is suitable for fitting point clouds to known object shapes, and a hybrid approach can combine more than one of the previous segmentation methods. As for machine learning clustering methods, k-means is more popular than hierarchical due to its simplicity, efficiency, and accuracy. Considering Grilli et al. [90] review of segmentation methods and their applicability, region-growing, model-fitting, and k-means clustering methods will be described in more detail.

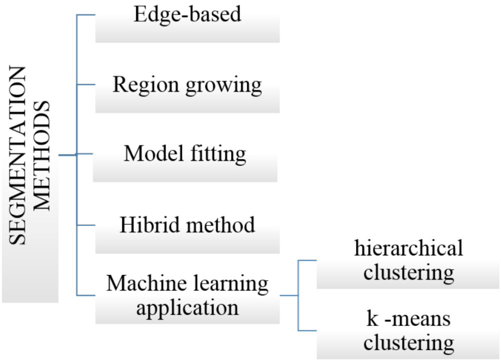


Figure 2.11: Synthetic representation of the segmentation methods. Figure retrieved from [90].

Region-growing

Region-growing is a technique that starts with a seed point and grows into a region by adding adjacent points that meet specific criteria, such as proximity and similarity. This method helps detect planar surfaces and separate objects from the background. A collection of region-growing algorithms is available in the Point Cloud Library [91]. Figure 2.12 shows the results of a region-growing segmentation implemented in the pcl::RegionGrowing class by Grilli et al. [90].

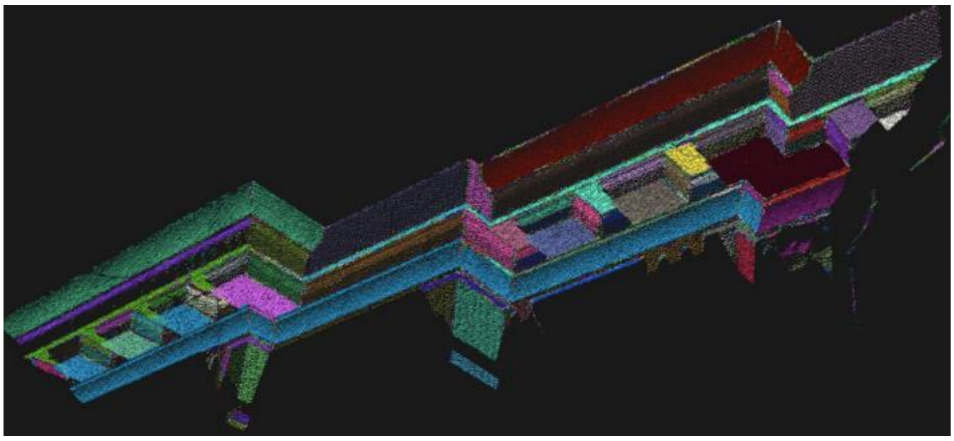


Figure 2.12: Point cloud segmented with a region-growing algorithm available in the Point Cloud Library. Figure retrieved from [90].

Model Fitting

Model fitting segmentation is a popular approach for segmenting point cloud data that contains geometric shapes like planes, cylinders, and spheres [92]. The method involves fitting geometric primitives onto the data and labelling the points that conform to the primitive shape as one segment. RANSAC algorithm is the most commonly used for model fitting-based segmentation due to its results quality and low running time. Furthermore, it may be used to fit models to data containing noise or outliers.

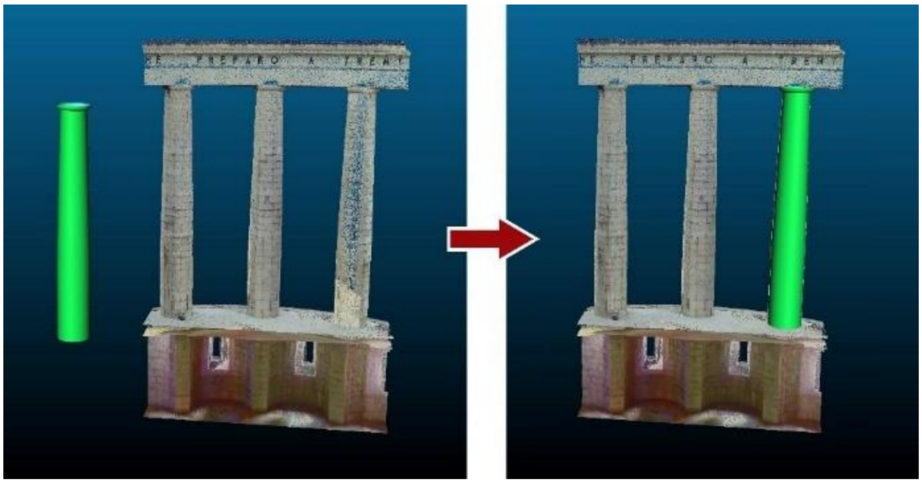
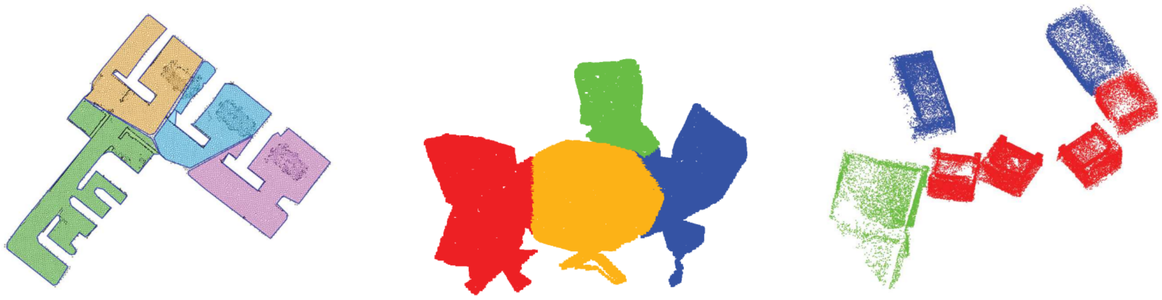


Figure 2.13: Example of a geometric primitive fitting on a point cloud. Figure retrieved from [90].

K-means Clustering

The K-means algorithm is an unsupervised learning method to cluster data points into K groups, where K is a predetermined number of clusters (Figure 2.14). The algorithm minimises the sum of squared distances between the data points and their assigned cluster centroids. It works by randomly selecting K points as the initial centroids, assigning each data point to the nearest centroid, recalculating the centroids based on the mean position of the assigned points, and repeating the process until the centroids no longer move or a predetermined number of iterations is reached. This method is very robust, but it is often a trade-off between quality and computational time. Furthermore, the number of clusters needs to be set beforehand. This algorithm is available in a collection of online libraries, for example, MATLAB [93].



(a) *Horizontally sliced floorplan segment (using only 2D X and Y coordinates of the points for clustering).*

(b) *k-means applied to a point cloud segment using 3D coordinates of the points.*

(c) *Use of k-means clustering to generate clusters for classification, with the post-classification result shown.*

Figure 2.14: Examples of k-means clustering applied to different point cloud data. Figure retrieved from [94].

This subsection introduces key algorithms for processing and analysing point cloud data. It is essential to stress the importance of incorporating a filtering technique to enhance data quality, followed by downsampling to reduce computational complexity. This data preparation paves the way for applying the discussed segmentation techniques to detect and track the motion of the supplier vessel's deck and compute the relative heave distance and speed information. Given the relatively large and flat nature of the ship's deck, employing a model-fitting approach is expected to yield favourable results. Additionally, leveraging data intensity readings from LiDAR and the Doppler velocity measurements from RADAR systems holds excellent potential to enhance the measuring system.

2.4. Literature Synthesis

This chapter offers a comprehensive literature review on the relative heave measurement topic. It begins with a broad overview of Ship-to-Ship cargo transfer operations, revealing a lack of existing literature on the subject, with Tørdal's work being the sole significant contribution. The review describes the current RHC systems, along with a critical research gap involving the proposal of a feeder-independent measuring solution for RHC.

Subsequently, section 2.2 discusses applicable sensor technologies for measuring relative motions, considering the characteristics of sensors available on the market. LiDAR, RADAR, and MRU sensors were found to be among the most suitable sensors for STS operations. Nevertheless, an analysis of existing research revealed an absence of measuring methods involving these advanced 3D sensors despite their considerable potential and sporadic observations for future research. Consequently, establishing a feeder-independent measuring system combining LiDAR-MRU, RADAR-MRU, or even LiDAR-RADAR-MRU was suggested as an essential avenue for research, addressing **Research Question 2**.

Lastly, section 2.3 explores data processing and analysis techniques essential for extracting relevant information about the supplier vessel's position from the obtained sensor data. It discusses various algorithms and potential data processing methods for measuring relative heave.

3 Relative Heave Modelling and Simulation

The following chapter explains the modelling and simulation methodology employed to address the relative heave problem in the context of STS operations, as initially introduced in chapter 1. First, a kinematic model is developed, addressing **Research Question 3**. The model is used to derive the exact relative heave distance and speed equations in terms of the ships' positions and attitudes. Second, the simulation workflow is presented, integrating MATLAB [95] and Unreal Engine (UE) [96] to generate LiDAR and RADAR data resembling the STS operation scenario, addressing **Research Question 4**.

3.1. Relative Heave Modelling

This section delves into the kinematic modelling of the relative heave, forming the fundamental understanding of the topic. The derived equations will be responsible for computing the reference values within the simulation. These mathematical expressions are incorporated into the Simulink block named `D_D_dot_Calculation`, presented in subsection 3.2.2. The section begins by introducing the geometric problem and its annotations. It then concisely explains the rigid-body kinematics and relevant mathematical notations used. After which, the derivation of the relative heave model kinematics is presented.

3.1.1. Geometric Definitions

This subsection provides a clear description of the scenario to be modelled and the annotations used in subsequent sections. Figure 3.1 provides an overview of the Ship-to-Ship cargo transfer operation, which will be modelled and discussed throughout this project. The coordinate systems are denoted using the bracket notation $\{\cdot\}$, where the specific coordinate system is represented by $\cdot$. This naming convention will frequently form the kinematic equations describing spatial body motions. The following section discusses the notation in more detail, so the reader is encouraged to review that section. Table 3.1 summarises the annotations depicted in Figure 3.1 and briefly describes each symbol.

Table 3.1: Description of the kinematic annotations presented in Figure 3.1

Symbol	Description
$\{b_1\}$	Coordinate system of ship 1.
$\{b_2\}$	Coordinate system of ship 2.
$\{g\}$	Global coordinate system.
$\{m\}$	Motion Reference Unit (MRU) mounted on ship body 1.
$\{s\}$	Exteroceptive sensor mounted on boom tip of ship body 1.
$\boldsymbol{p_t}$	The crane tip point where the Wire Exit Point (WEP) is located
$\boldsymbol{n_w}$	The unit vector following the wire connecting the crane and the hook.
$\boldsymbol{p_d}$	The point where a line in the direction of $\boldsymbol{n_w}$ intersects the ship 2 deck.
D	Distance between points $\boldsymbol{p_t}$ and $\boldsymbol{p_d}$

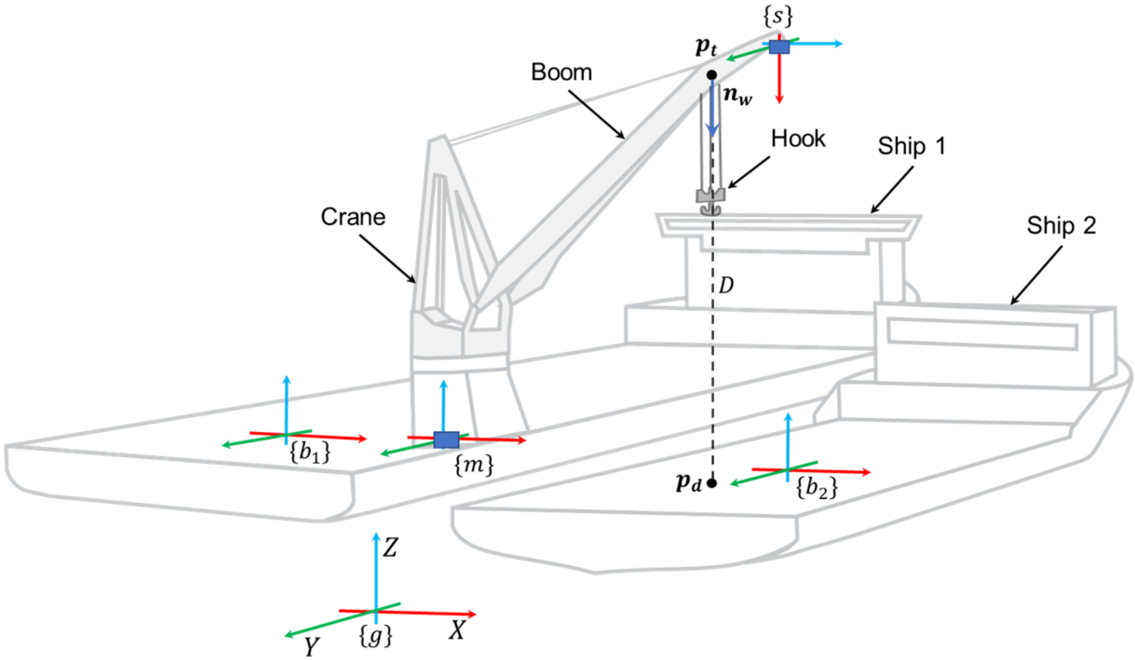


Figure 3.1: Illustration of the various coordinate systems used in Ship-to-Ship cargo transfer operations. Ship 1, alternatively known as the crane vessel/ship, and Ship 2, also referred to as the supply/feeder vessel/ship, are involved in the process.

3.1.2. Rigid-Body Kinematics and Mathematical Notation

This section discusses the fundamental notation used to describe the geometric relationship between vectors, coordinate systems, and bodies. The notation and kinematic formulations are based on the work presented by Fossen [20] and used by Tørdal [11] to model Ship-to-Ship operations. However, two minor modifications have been implemented: the adoption of left-handed coordinate systems and the alignment of the z-axis of the global reference frame in the opposite direction to gravity, to ensure compatibility with the simulation environment.

Rather than delving into the intricate details of this notation, which can be found in Fossen's work[20], a straightforward example of a moving rigid body will be provided to illustrate the notation employed in this research.

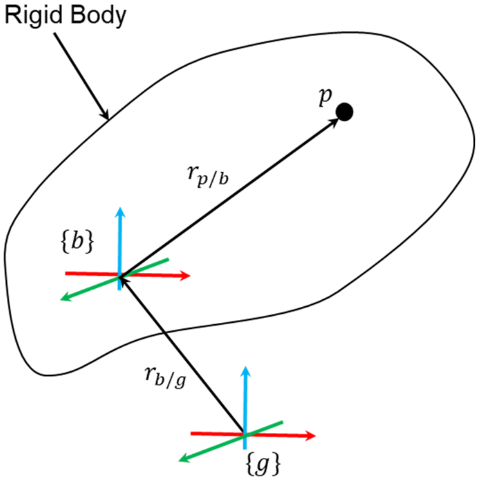


Figure 3.2: Representation of a point p located on a rigid body represented by coordinate system $\{b\}$ which is moving relative to a static global coordinate system $\{g\}$.

As seen in Figure 3.2, the position between the body coordinate $\{b\}$, and the global $\{g\}$ is defined using vector $r_{b/g}$, and the orientation of $\{b\}$ relative to $\{g\}$ is given by a rotation matrix $\boldsymbol{R}_b^g$. The subscript of vector $\boldsymbol{r}_{b/g}$ defines the direction of the vector, hence $\boldsymbol{r}_{b/g} = -\boldsymbol{r}_{g/b}$. In addition, the vector superscript is used to define which coordinate the vector is given in (i.e. if the vector is given in $\{g\}$, the following superscript is used $\boldsymbol{r}^{g}_{b/g}$). See the following expression for a better understanding of the notation.

$$\boldsymbol{r}^{g}_{b/g} = \boldsymbol{R}^{g}_{b}\boldsymbol{r}^{b}_{b/g} = \boldsymbol{R}^{g}_{b}(-\boldsymbol{r}^{b}_{g/b}), \quad \boldsymbol{R}^{g}_{b} \in SO(3) \tag{3.1}$$

Where $SO(3)$ represents the special orthogonal group in 3D, also known as rotation group. Hence, the inverse of a rotation matrix is equal to its transpose.

$$\boldsymbol{R}^{b}_{g} = (\boldsymbol{R}^{g}_{b})^{-1} = (\boldsymbol{R}^{g}_{b})^{T} \tag{3.2}$$

Using this notation and considering point $\boldsymbol{p}$ is fixed to the rigid-body $\{b\}$ (i.e. $\dot{\boldsymbol{r}}^{b}_{p/b} = 0$), we have that the position and velocity of point $\boldsymbol{p}$ relative to $\{g\}$ is given by:

$$\boldsymbol{r}^{g}_{p/g} = \boldsymbol{r}^{g}_{b/g} + \boldsymbol{R}^{g}_{b}\boldsymbol{r}^{b}_{p/b} \tag{3.3}$$

$$\dot{\boldsymbol{r}}^{g}_{p/g} = \dot{\boldsymbol{r}}^{g}_{b/g} + \dot{\boldsymbol{R}}^{g}_{b}\boldsymbol{r}^{b}_{p/b} \tag{3.4}$$

The rotation matrix $\boldsymbol{R}^{g}_{b}$ is commonly represented as a combination of rotations around the Euler angles $\boldsymbol{\Theta}gb = [\phi, \theta, \psi]^{T}$, which in this case describe the body's attitude relative to the global reference frame. To calculate $\dot{\boldsymbol{R}}^{g}_{b}$, the partial derivatives of $\boldsymbol{R}^{g}_{b}$ with respect to the Euler angles (ϕ, θ, ψ) are multiplied by the body fixed angular velocities $\boldsymbol{w}^{\boldsymbol{g}}_{\boldsymbol{b/g}} = [\dot{\phi}, \dot{\theta}, \dot{\psi}]^{T}$ of body $\{b\}$ relative to ground $\{g\}$ given in the body coordinates $\{b\}$. This relationship can be mathematically expressed as follows:

$$\dot{\boldsymbol{R}}^{g}_{b} = \frac{\partial \boldsymbol{R}^{g}_{b}}{\partial \phi}\dot{\phi} + \frac{\partial \boldsymbol{R}^{g}_{b}}{\partial \theta}\dot{\theta} + \frac{\partial \boldsymbol{R}^{g}_{b}}{\partial \psi}\dot{\psi} \tag{3.5}$$

Throughout the thesis, the vectors and variables are considered time-dependent. However, the time dependency is omitted to save space and avoid equations spanning multiple lines. For example:

$$\boldsymbol{x(t)} \to \boldsymbol{x} \tag{3.6}$$

It is worth mentioning that if a vector or parameter is time-invariant, it is clarified in the text, and the corresponding time derivative is zero.

3.1.3. Relative Heave Kinematic Model

This section outlines the steps taken to develop the mathematical model for computing the relative heave distance and speed between two ships. The model considers all six degrees of freedom: surge, sway, heave, roll, pitch, and yaw. Note that these motions are assumed to have zero mean over time relative to the fixed global coordinate system, denoted as $\{g\}$, with its z-axis aligned opposite to the gravity vector.

Before embarking on the detailed modelling of the problem, the simplifications made can be summarised as follows:

1. The crane and supplier vessels are modelled as independent rigid bodies.
2. The swinging motions of the load are neglected.
3. The wire rope is assumed to be perfectly rigid and without any elasticity.
4. The kinematics of the crane are not taken into account.

It is important to remark that these assumptions align with current industry practices. In maritime applications, it is common to model ships as rigid bodies. Additionally, neglecting the wire rope's swinging motions and elastic behaviour is a reasonable approximation for the specific application.

Based on these assumptions, the vertical distance between the hook and the ship deck can be determined by subtracting the length of the paid-out wire from the value of D, depicted in Figure 3.3. Hence, in this thesis, relative heave distance (D) corresponds to the distance between the crane tip ($\boldsymbol{p_t}$) and the point of intersection between the unit vector ($\boldsymbol{n_w}$) and the deck plane of ship 2, ($\boldsymbol{p_d}$).

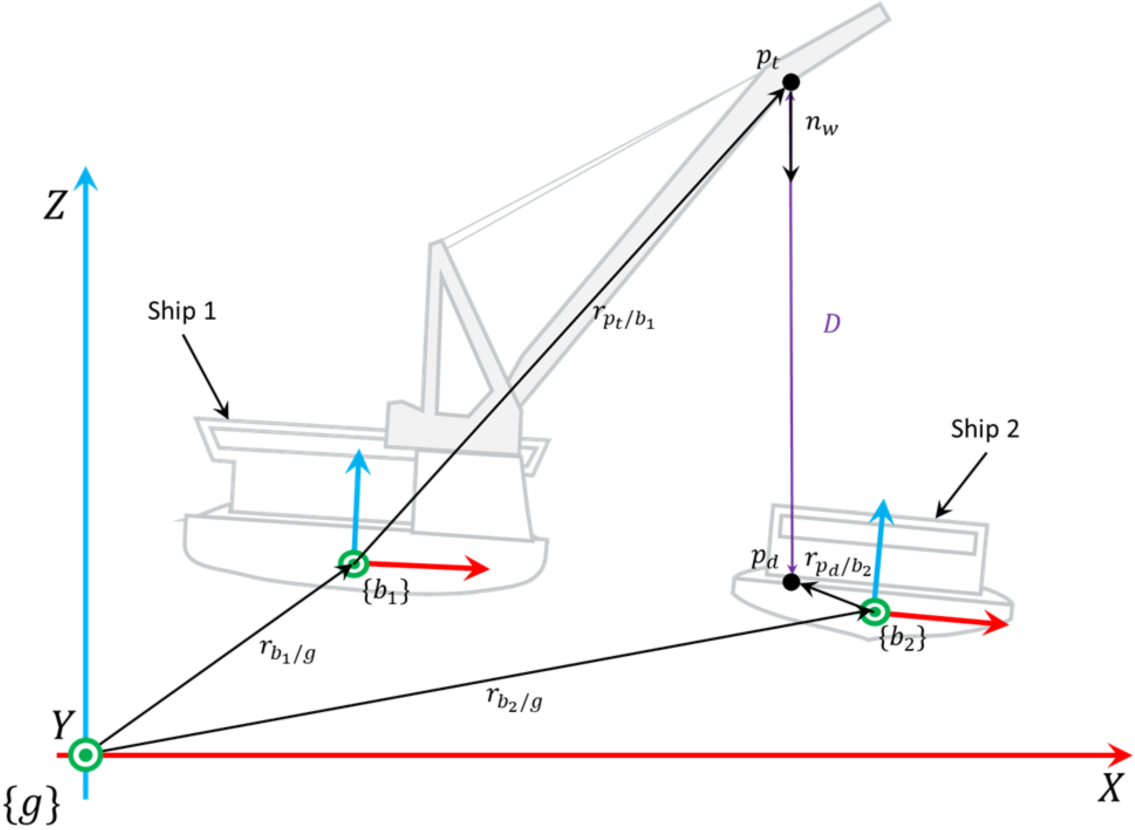


Figure 3.3: Kinematics used to describe relative heave in Ship-to-Ship cargo transfer operations.

Based on the previous Figure 3.3, the following mathematical expression can be derived:

$$\boldsymbol{r}^g_{p_t/g} + \boldsymbol{n}^g_w \cdot D = \boldsymbol{r}^g_{p_d/g} \tag{3.7}$$

Where, $\boldsymbol{r}^g_{p_t/g} = [x^g_{p_t}, y^g_{p_t}, z^g_{p_t}]^T$ and $\boldsymbol{r}^g_{p_d/g} = [x^g_{p_d}, y^g_{p_d}, z^g_{p_d}]^T$ denote the coordinates of $\boldsymbol{p_t}$ and $\boldsymbol{p_d}$ in the global coordinate system, respectively. Additionally, the unit vector $\boldsymbol{n}^g_w$ represents the direction of the wire connecting the crane to the load. Assuming no load-swinging motions occur, this vector remains constantly aligned with gravity and can be described as follows:

$$\boldsymbol{n}^g_w = \begin{bmatrix} 0 \\ 0 \\ -1 \end{bmatrix} \tag{3.8}$$

Combining Equation 3.7 with Equation 3.8 and rearranging the terms, the following expression is obtained:

$$\boldsymbol{r}^g_{p_t/g} - \boldsymbol{r}^g_{p_d/g} = \begin{bmatrix} 0 \\ 0 \\ D \end{bmatrix} \tag{3.9}$$

In order to ascertain the value of D, the expressions for $\boldsymbol{r}^g_{p_t/g}$ and $\boldsymbol{r}^g_{p_d/g}$ need to be established based on the known variables (ship motions set in Simulink). By examining Figure 3.3, these vectors can be formulated as follows:

$$\boldsymbol{r}^g_{p_t/g} = \boldsymbol{r}^g_{b_1/g} + \boldsymbol{R}^g_{b_1} \boldsymbol{r}^{b_1}_{p_t/b_1} \tag{3.10}$$

$$\boldsymbol{r}^g_{p_d/g} = \boldsymbol{r}^g_{b_2/g} + \boldsymbol{R}^g_{b_2} \boldsymbol{r}^{b_2}_{p_d/b_2} \tag{3.11}$$

Finally, substituting eq. (3.10) and (3.11) on eq. (3.9),

$$\boldsymbol{r}^{g}_{b_1/g} + \boldsymbol{R}^{g}_{b_1}\boldsymbol{r}^{b_1}_{p_t/b_1} - \boldsymbol{r}^{g}_{b_2/g} - \boldsymbol{R}^{g}_{b_2}\boldsymbol{r}^{b_2}_{p_d/b_2} = \begin{bmatrix} 0 \\ 0 \\ D \end{bmatrix} \tag{3.12}$$

To accurately determine D, it is necessary to fully define the position $\boldsymbol{r}$ and attitude $\boldsymbol{\Theta}$ of the ships in space. The subsequent vectors provide a comprehensive description of the position and attitude of ship 1 and ship 2 relative to the global coordinate system $\{g\}$:

$$\boldsymbol{r}^{g}_{b_1/g} = \begin{bmatrix} x_{b_1} \\ y_{b_1} \\ z_{b_1} \end{bmatrix}, \quad \boldsymbol{\Theta}_{gb_1} = \begin{bmatrix} \phi_{b_1} \\ \theta_{b_1} \\ \psi_{b_1} \end{bmatrix}, \quad \boldsymbol{r}^{g}_{b_2/g} = \begin{bmatrix} x_{b_2} \\ y_{b_2} \\ z_{b_2} \end{bmatrix}, \quad \boldsymbol{\Theta}_{gb_2} = \begin{bmatrix} \phi_{b_2} \\ \theta_{b_2} \\ \psi_{b_2} \end{bmatrix} \tag{3.13}$$

Where $\boldsymbol{r}^{g}_{b_1/g}$ and $\boldsymbol{r}^{g}_{b_2/g}$ represent Ship 1 and Ship 2 positions, respectively. Moreover, the Euler angles, denoted by $\boldsymbol{\Theta}_{gb_1}$ and $\boldsymbol{\Theta}_{gb_2}$, describe Ship 1 and Ship 2 attitude relative to the global coordinate system, respectively.

Given the Euler angles describing a ship body attitude ($\boldsymbol{\Theta}_{gb}$), the combined rotation matrix ($\boldsymbol{R}^{g}_{b}$) that relates a vector in the body-fixed frame to a vector in the global reference frame can be obtained. Following the convention employed by Unreal Engine, which follows the order of extrinsic rotations with a ZYX-angle sequence, the rotation matrix can be expressed as:

$$\boldsymbol{R}^{g}_{b}(\boldsymbol{\Theta}_{gb}) = \boldsymbol{R}_z(\psi)\boldsymbol{R}_y(\theta)\boldsymbol{R}_x(\phi) \tag{3.14}$$

Where $\boldsymbol{R}_x$, $\boldsymbol{R}_y$ and $\boldsymbol{R}_z$ are the well established rotation matrices used around x-, y- and z-axis. Considering that the coordinate systems used are left-handed and the Euler angles describe the ship rotations relative to an external fixed frame $\{g\}$ (extrinsic), the three elemental rotation matrices can be represented as:

$$\boldsymbol{R}_x(\phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos(\phi) & sin(\phi) \\ 0 & -sin(\phi) & cos(\phi) \end{bmatrix}, \ \boldsymbol{R}_y(\theta) = \begin{bmatrix} cos(\theta) & 0 & -sin(\theta) \\ 0 & 1 & 0 \\ sin(\theta) & 0 & cos(\theta) \end{bmatrix}, \ \boldsymbol{R}_z(\psi) = \begin{bmatrix} cos(\psi) & sin(\psi) & 0 \\ -sin(\psi) & cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.15}$$

Now, combining eq. (3.14) and (3.15). The resulting matrix can be computed as:

$$\boldsymbol{R}^{g}_{b}(\boldsymbol{\Theta}_{gb}) = \begin{bmatrix} cos(\psi)cos(\theta) & cos(\phi)sin(\psi) + cos(\psi)sin(\phi)sin(\theta) & sin(\phi)sin(\psi) - cos(\phi)cos(\psi)sin(\theta) \\ -cos(\theta)sin(\psi) & cos(\phi)cos(\psi) - sin(\phi)sin(\psi)sin(\theta) & cos(\psi)sin(\phi) + cos(\phi)sin(\psi)sin(\theta) \\ sin(\theta) & -cos(\theta)sin(\phi) & cos(\phi)cos(\theta) \end{bmatrix} \tag{3.16}$$

Equations 3.13 and 3.16 define most of the terms in Equation 3.12, but two terms remain undefined. One of them is $\boldsymbol{r}^{b_1}_{p_t/b_1}$, representing the position of the crane tip $\boldsymbol{p_t}$ relative to the body-fixed frame $\{b_1\}$.

It is important to note that this vector is affected by the crane's geometry configuration and the current slew and boom angle during operation. However, for this research, the crane's kinematics are not within the scope. Therefore, the crane is assumed to be static relative to the ship frame, resulting in the following vectors:

$$\boldsymbol{r}^{b_1}_{p_t/b_1} = \begin{bmatrix} x_{p_t} \\ y_{p_t} \\ z_{p_t} \end{bmatrix}, \quad \dot{\boldsymbol{r}}^{b_1}_{p_t/b_1} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \tag{3.17}$$

The remaining vector $\boldsymbol{r}^{b_2}_{p_d/b_2}$ represents the position of $\boldsymbol{p_d}$ with respect to $\{b_2\}$. The value of this vector depends on the positions of $\boldsymbol{p_t}$ and $\{b_2\}$. Although $\boldsymbol{r}^{b_2}_{p_d/b_2}$ is initially unknown, it can be determined using other known system vectors. It is important to note that the position $\boldsymbol{r}^{b_2}_{p_d/b_2}$ changes over time. However, the distance of the ship 2 deck along the z-axis relative to $\{b_2\}$ remains constant. As a result, z_{p_d} remains unchanged over time, leading to the following vectors:

$$\boldsymbol{r}^{b_2}_{p_d/b_2} = \begin{bmatrix} x_{p_d} \\ y_{p_d} \\ z_{p_d} \end{bmatrix}, \quad \dot{\boldsymbol{r}}^{b_2}_{p_d/b_2} = \begin{bmatrix} \dot{x}_{p_d} \\ \dot{y}_{p_d} \\ 0 \end{bmatrix} \tag{3.18}$$

By replacing the vector notation utilised in the system of equations (3.12), and expanding the system in its entirety, the subsequent expression is obtained:

$$\begin{aligned}&\begin{bmatrix} x_{b_1} \\ y_{b_1} \\ z_{b_1} \end{bmatrix} + \begin{bmatrix} c_{\psi_{b_1}} c_{\theta_{b_1}} & c_{\phi_{b_1}} s_{\psi_{b_1}} + c_{\psi_{b_1}} s_{\phi_{b_1}} s_{\theta_{b_1}} & s_{\phi_{b_1}} s_{\psi_{b_1}} - c_{\phi_{b_1}} c_{\psi_{b_1}} s_{\theta_{b_1}} \\ -c_{\theta_{b_1}} s_{\psi_{b_1}} & c_{\phi_{b_1}} c_{\psi_{b_1}} - s_{\phi_{b_1}} s_{\psi_{b_1}} s_{\theta_{b_1}} & c_{\psi_{b_1}} s_{\phi_{b_1}} + c_{\phi_{b_1}} s_{\psi_{b_1}} s_{\theta_{b_1}} \\ s_{\theta_{b_1}} & -c_{\theta_{b_1}} s_{\phi_{b_1}} & c_{\phi_{b_1}} c_{\theta_{b_1}} \end{bmatrix} \begin{bmatrix} x_{p_t} \\ y_{p_t} \\ z_{p_t} \end{bmatrix} - \\ &\quad \begin{bmatrix} x_{b_2} \\ y_{b_2} \\ z_{b_2} \end{bmatrix} - \begin{bmatrix} c_{\psi_{b_2}} c_{\theta_{b_2}} & c_{\phi_{b_2}} s_{\psi_{b_2}} + c_{\psi_{b_2}} s_{\phi_{b_2}} s_{\theta_{b_2}} & s_{\phi_{b_2}} s_{\psi_{b_2}} - c_{\phi_{b_2}} c_{\psi_{b_2}} s_{\theta_{b_2}} \\ -c_{\theta_{b_2}} s_{\psi_{b_2}} & c_{\phi_{b_2}} c_{\psi_{b_2}} - s_{\phi_{b_2}} s_{\psi_{b_2}} s_{\theta_{b_2}} & c_{\psi_{b_2}} s_{\phi_{b_2}} + c_{\phi_{b_2}} s_{\psi_{b_2}} s_{\theta_{b_2}} \\ s_{\theta_{b_2}} & -c_{\theta_{b_2}} s_{\phi_{b_2}} & c_{\phi_{b_2}} c_{\theta_{b_2}} \end{bmatrix} \begin{bmatrix} x_{p_d} \\ y_{p_d} \\ z_{p_d} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ D \end{bmatrix}\end{aligned} \tag{3.19}$$

Finally, having a system of three equations (3.19) and three unknowns, it is possible to determine D. Solving eq. 3.19 for D along with the unknowns x_{p_d} and y_{p_d} mentioned in equation (3.18), the relative heave can be computed by the following expression:

$$D = f(\boldsymbol{r}^g_{b_1}, \boldsymbol{\Theta}_{gb_1}, \boldsymbol{r}^g_{b_2}, \boldsymbol{\Theta}_{gb_2}, \boldsymbol{r}^{b_1}_{pt}, z_{pd}) \;, \quad z_{pd} = const. \tag{3.20}$$

Furthermore, taking the time derivative of eq. (3.12), the following expression is obtained for the relative heave rate of change:

$$\dot{\boldsymbol{r}}^g_{b_1/g} + (\dot{\boldsymbol{R}}^g_{b_1} \boldsymbol{r}^{b_1}_{p_t/b_1} + \boldsymbol{R}^g_{b_1} \dot{\boldsymbol{r}}^{b_1}_{p_t/b_1}) - \dot{\boldsymbol{r}}^g_{b_2/g} - (\dot{\boldsymbol{R}}^g_{b_2} \boldsymbol{r}^{b_2}_{p_d/b_2} + \boldsymbol{R}^g_{b_2} \dot{\boldsymbol{r}}^{b_2}_{p_d/b_2}) = \begin{bmatrix} 0 \\ 0 \\ \dot{D} \end{bmatrix} \tag{3.21}$$

Using Equation 3.17, the previous Equation 3.21 can be simplified to:

$$\dot{\boldsymbol{r}}^g_{b_1/g} + \dot{\boldsymbol{R}}^g_{b_1} \boldsymbol{r}^{b_1}_{p_t/b_1} - \dot{\boldsymbol{r}}^g_{b_2/g} - (\dot{\boldsymbol{R}}^g_{b_2} \boldsymbol{r}^{b_2}_{p_d/b_2} + \boldsymbol{R}^g_{b_2} \dot{\boldsymbol{r}}^{b_2}_{p_d/b_2}) = \begin{bmatrix} 0 \\ 0 \\ \dot{D} \end{bmatrix} \tag{3.22}$$

Where, the time derivatives of the position and attitude vectors, representing the linear velocity $(\dot{\boldsymbol{r}})$ and angular velocity $(\boldsymbol{w})$ of both ships with respect to $\{g\}$, can be expressed as follows:

$$\dot{\boldsymbol{r}}^g_{b_1/g} = \begin{bmatrix} \dot{x}_{b_1} \\ \dot{y}_{b_1} \\ \dot{z}_{b_1} \end{bmatrix} , \quad \boldsymbol{w}_{gb_1} = \begin{bmatrix} \dot{\phi}_{b_1} \\ \dot{\theta}_{b_1} \\ \dot{\psi}_{b_1} \end{bmatrix} , \quad \dot{\boldsymbol{r}}^g_{b_2/g} = \begin{bmatrix} \dot{x}_{b_2} \\ \dot{y}_{b_2} \\ \dot{z}_{b_2} \end{bmatrix} , \quad \boldsymbol{w}_{gb_2} = \begin{bmatrix} \dot{\phi}_{b_2} \\ \dot{\theta}_{b_2} \\ \dot{\psi}_{b_2} \end{bmatrix} \tag{3.23}$$

By substituting the solution obtained from Equation 3.12 for x_{p_d} and y_{p_d}, and solving the system of equations derived from Equation 3.22 for $\dot{D}$, $\dot{x_{p_d}}$, and $\dot{y_{p_d}}$, the value of $\dot{D}$ can be determined based solely on the known variables. The following expression allows for the computation of the relative heave speed:

$$\dot{D} = g(\dot{\boldsymbol{r}}^g_{b_1}, \boldsymbol{w}_{gb_1}, \boldsymbol{\Theta}_{gb_1}, \dot{\boldsymbol{r}}^g_{b_2}, \boldsymbol{w}_{gb_2}, \boldsymbol{\Theta}_{gb_1}, z_{pd}) \tag{3.24}$$

The functions f and g have been excluded from this section due to their substantial size. However, function f is provided in the Appendix C for reference.

3.2. Simulation Implementation

This research aims to develop effective methods for measuring relative heave during Ship-to-Ship cargo transfer processes. It involves extracting desired measurements and estimates from 3D RADAR and LiDAR raw sensor data. Hence, obtaining this data is the initial step before exploring various data processing and analysis techniques. There are two primary approaches to acquire this data: employing actual sensors within an experimental setup or utilising sensor models within a simulation environment.

Both approaches have their advantages and disadvantages. Real data offers a higher level of representation if the experimental setup accurately reflects the problem under study. However, it is often more expensive, and, in the context of this thesis, the necessary tools were not initially available. Given these constraints, developing a simulation workflow to generate representative simulated data for Ship-to-Ship cargo transfer scenarios emerged as the most logical choice. This decision enables swift development and implementation of different methodologies within a controlled simulation environment.

In order to simulate a complex multi-body system in 3D space and accurately generate RADAR and LiDAR point clouds, reliable simulation platforms such as Gazebo [97] and Simulink-MATLAB [98] offer significant advantages. These platforms offer comprehensive 3D simulation tools and sensor models, enabling the testing of various sensors, configurations, and algorithms. Moreover, they facilitate efficient exploration and evaluation of various scenarios and methodologies.

This thesis, in particular, combines MATLAB 2022b and Unreal Engine 4.26 to create an adequate simulation for STS operations. Although primarily employed in autonomous driving, robotics, and Unmanned Aerial Vehicles (UAV) applications, the 3D simulation software tools offer possibilities beyond these domains. The decision to adopt this approach was strongly influenced by a compelling example presented by MathWorks [99], highlighting its effectiveness and adaptability. By leveraging the strengths of these platforms, a comprehensive simulation environment was created to fulfil the research objectives. Importantly, to the authors' knowledge, no simulation environment had been specifically designed for this application prior to this study. Figure 3.4 provides a schematic overview of the general concept behind the developed simulation workflow.

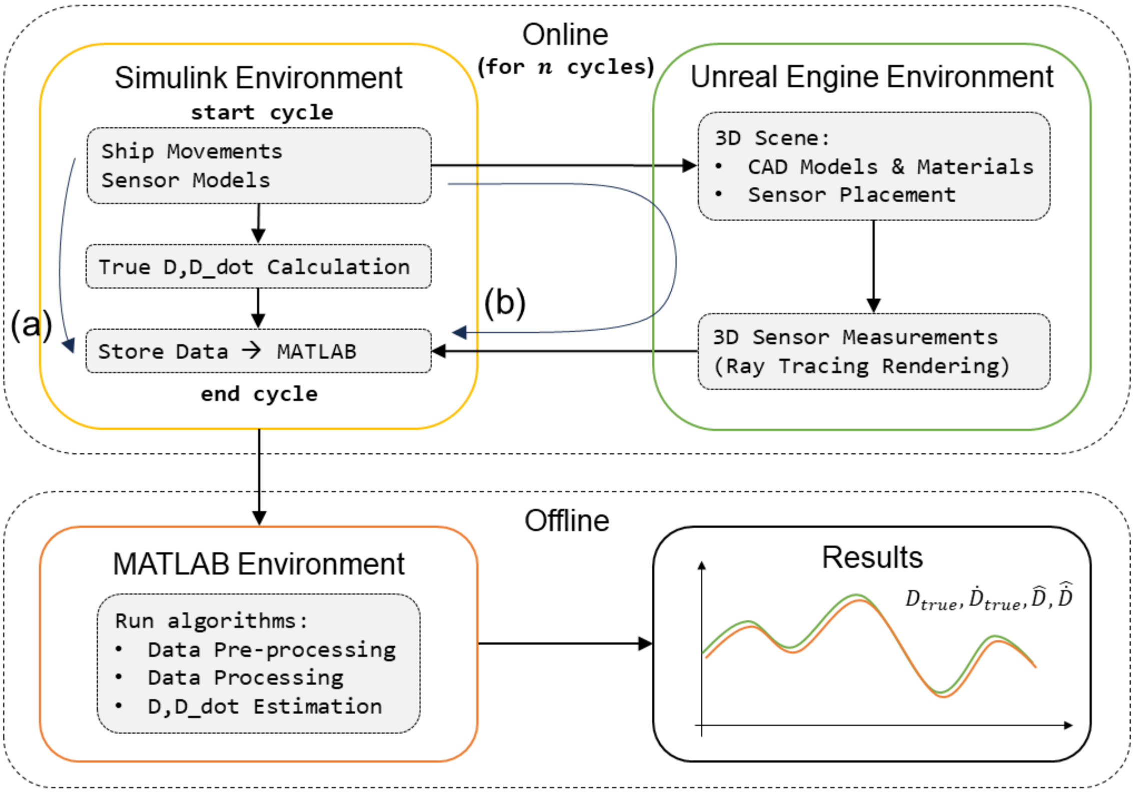


Figure 3.4: Overview of the implemented data generation and processing workflow using Simulink, Unreal Engine (UE), and MATLAB. The process involves two concurrent cycles: (a) Capturing ground truth values of relative heave and (b) Simulating 3D point cloud data from RADAR and LiDAR. Simulink and UE are utilised for online data generation and storage, while MATLAB is employed for offline data processing.

The software structure, depicted in Figure 3.4, outlines the flow from data acquisition to the processing and analysis of results. During the simulation, data is generated and stored in real-time over a user-defined period. The Simulink model continuously generates the positions and attitudes of each ship, emulating their responses to sea conditions. These ship positions and attitudes are then transmitted to and enacted within Unreal Engine, causing the 3D CAD models to move accordingly for each time frame.

Sensor models, implemented in Simulink and placed at the boom tip of the Unreal Engine environment, capture the ship's movements using UE's Ray Tracing Rendering technology. The data collected by these sensors is subsequently stored as a time-series MATLAB file. Additionally, the ground truth relative heave values, acquired from the ships' kinematics, are also stored in parallel.

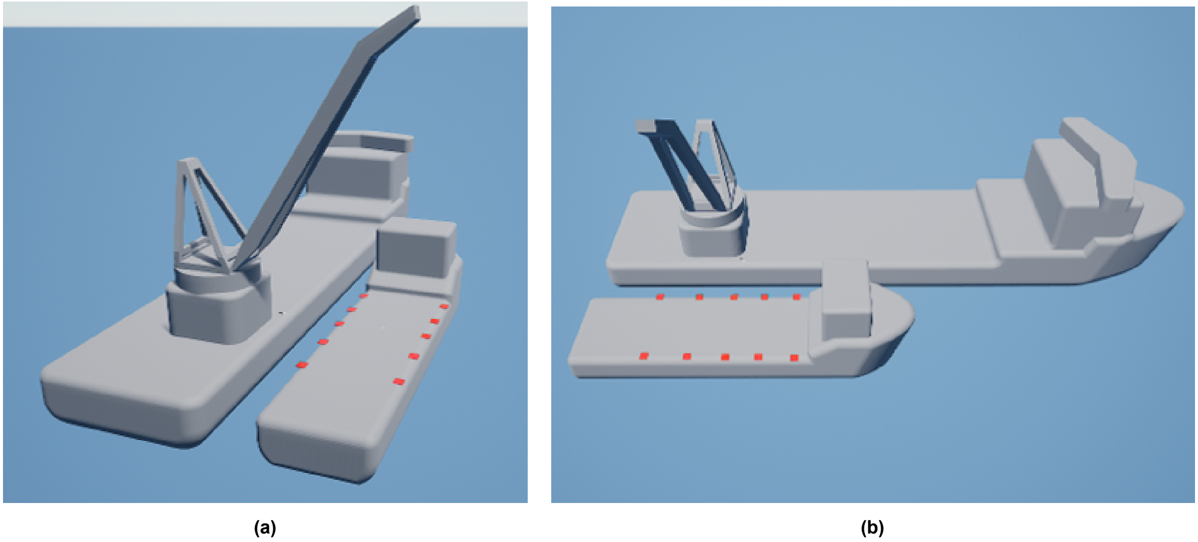
To conclude, the data processing and analysis algorithms can be developed and evaluated using both the ground truth values and the generated sensor data for each time frame. This phase occurs within MATLAB, as depicted in the preceding Figure 3.4. The following subsections will provide a more detailed explanation of the Simulink and Unreal Engine implementations for data acquisition.

3.2.1. Unreal Engine Implementation

Unreal Engine, developed by Epic Games, is widely recognised for its advanced features in creating immersive virtual environments and simulations. In the context of simulating Ship-to-Ship cargo transfers using LiDAR and RADAR sensors, Unreal Engine offers notable advantages.

Firstly, UE offers a user-friendly interface that simplifies the import of CAD models, allowing for the examination of different ship types and scenarios. Moreover, it supports the application of diverse materials and characteristics to these models, resulting in a more realistic simulation. Secondly, the seamless integration of real-time body kinematics within the virtual environment enables accurate representation of ship movements. Finally, the great ray tracing software of UE enhances the precision of the sensor models used. This advanced graphical feature guarantees that the LiDAR and RADAR data accurately represent the surrounding environment.

This study used Solid-Edge CAD software to model the crane and supplier vessels. The dimensions of the ships were acquired from "Les Alizés," an installation vessel owned by Jan de Nul [100], as well as a smaller feeder vessel. Subsequently, these models were imported into the virtual environment of UE. Figure 3.5 illustrates the arrangement of the Ship-to-Ship cargo transfer scenario.


(a)

(b)

Figure 3.5: STS scenario setup in UE. Two vessels are positioned side-by-side for cargo transfer, with an empty supplier vessel and a crane vessel ready for operation.

RADAR and LiDAR data can be affected by several factors, such as the object's distance, material composition, orientation, and reflectivity. In this simulation, both ship bodies are constructed with rugged metallic steel, resulting in a greyish appearance. Additionally, red markers made of highly reflective material are positioned along the edges of the supplier vessel to be leveraged in future data processing and analysis techniques.

The preceding simplified scenario, featuring an empty supplier vessel, is employed for the algorithms verification stage. However, a secondary scenario involving a fully loaded supply ship is implemented to assess the algorithms under more realistic deck conditions at the evaluation stage (Figure 3.6).

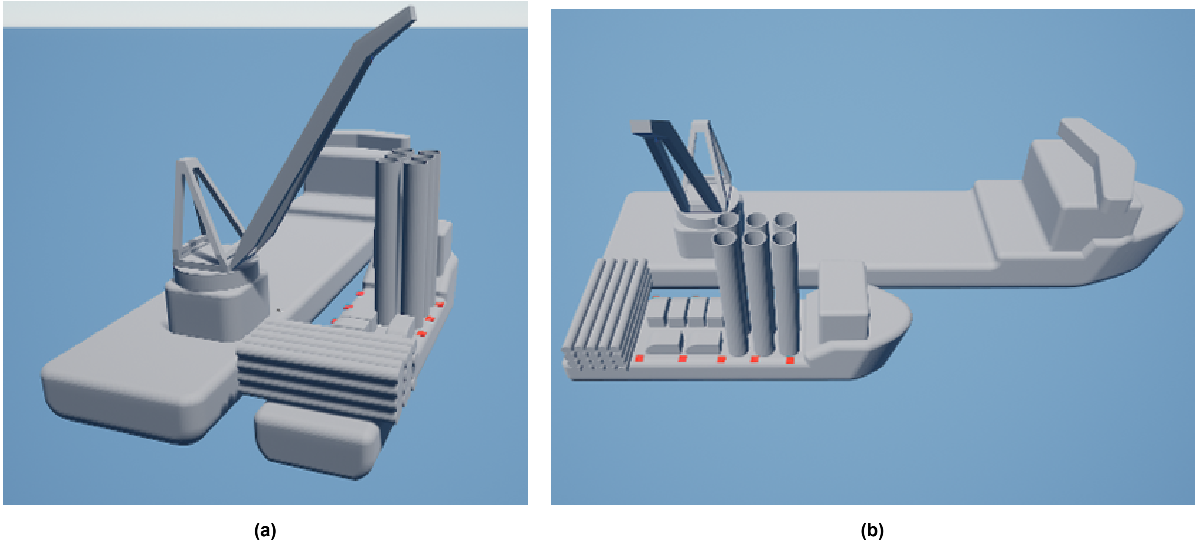

(a) **(b)**

Figure 3.6: STS scenario setup in UE. Two vessels are positioned side-by-side for cargo transfer, with a loaded supplier vessel and a crane vessel ready for operation.

The implemented Unreal Engine scenario provides the 3D simulation environment, wherein the ship models are subsequently moved according to the motions generated by the Simulink model. The following subsection 3.2.2 presents the detailed Simulink Model used.

To illustrate the movement of the ships during the simulation, Figure 3.7 depicts three subsequent time frames of the UE simulation display window.

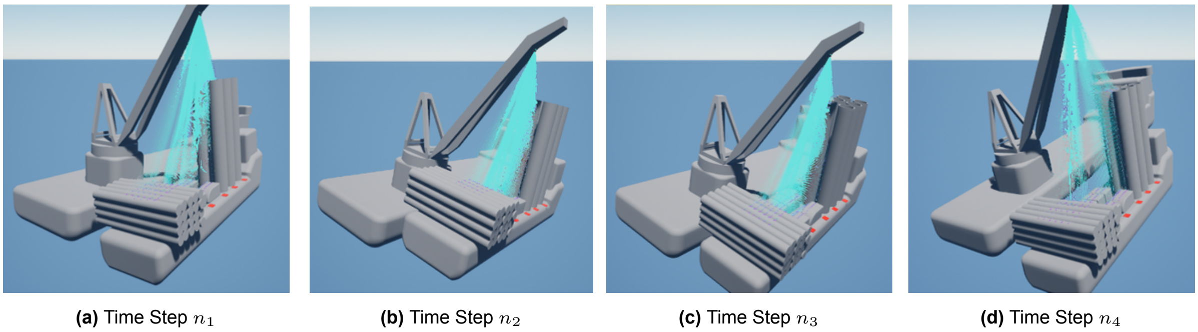

(a) Time Step n_1 **(b)** Time Step n_2 **(c)** Time Step n_3 **(d)** Time Step n_4

Figure 3.7: Temporal evolution of the ship bodies in Unreal Engine (UE) with $n_1 < n_2 < n_3$. The ships' positions and attitudes have been exaggerated to ease readers' understanding of the simulation behaviour.

Figure 3.7 also helps understand how ray tracing works to generate sensor measurements. Rays are created based on the FOV and specifications of the 3D sensor. When these rays (blue) intersect with another body, the distance relative to the sensor in the ray's direction is recorded. This data is then stored in the format of a point cloud containing the spatial location of the ray hits relative to the sensor frame. Examples of such point clouds are illustrated in Figure 3.10 and Figure 3.11.

3.2.2. Simulink Implementation

Simulink is a visual programming environment and simulation tool developed by MathWorks. It is widely used for modelling, simulating, and analysing dynamic systems. As explained before, Simulink is a crucial part of the implemented simulation workflow. The resulting Simulink block diagram is depicted in Figure 3.8.

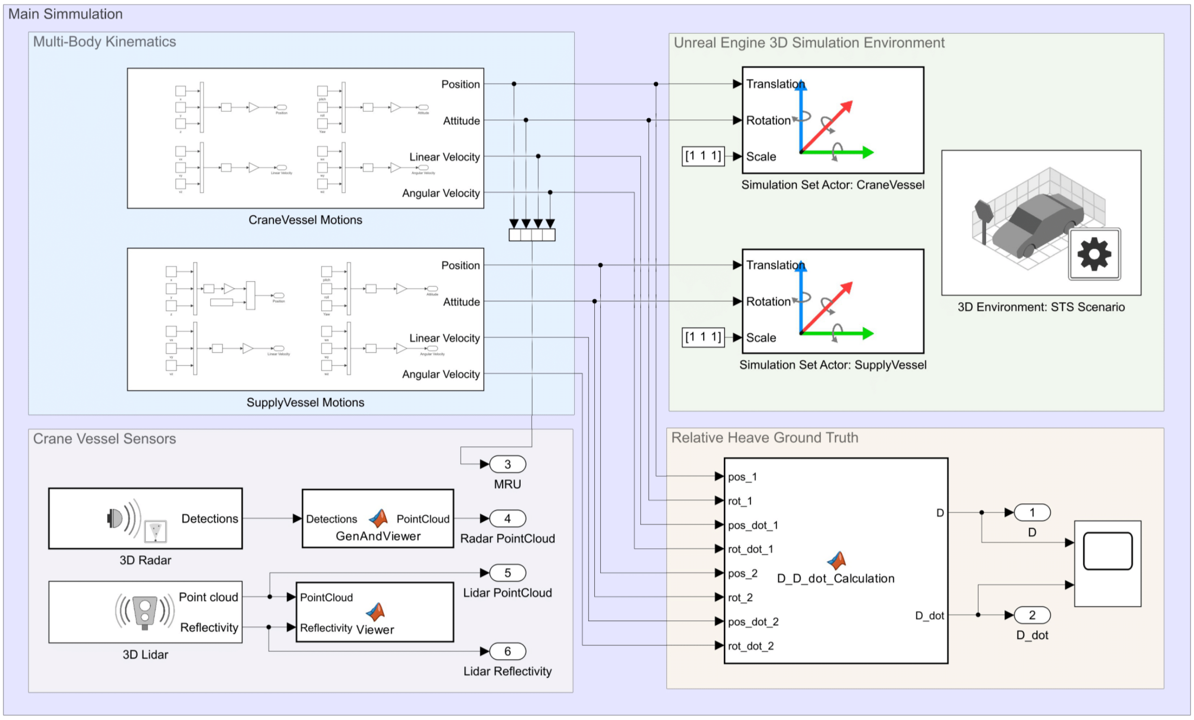


Figure 3.8: Simulink model for Ship-to-Ship kinematics simulation and 3D sensor data generation.

Figure 3.8 illustrates four main subsystems that are important to describe. The “Multi-body Kinematics” subsystem, located in the top left, generates the ship motions for both the crane and supplier vessels, considering their six DOFs. Stochastic wave forces and accurate ship dynamics can effectively model these intricate ship movements, as described in Fossen’s handbook [20]. However, for the current application, the influence of irregular waves on the development of the measuring methodologies is negligible. Thus, this study assumes regular sinusoidal motions for both ship bodies in all 6DOFs. It is worth mentioning that the ship’s motions can be readily replaced if, at a later stage, more sophisticated dynamics are required.

In the top right corner, the “Unreal Engine 3D Simulation Environment” subsystem relays the Simulink-generated motions to the rigid bodies within its environment. To ensure seamless communication between the software, it is crucial for the body and sensor tags in both environments to match. For detailed steps and requirements for this communication to function correctly, please consult the Simulink documentation [98].

The “Crane Vessel Sensors” subsystem at the bottom left captures the data generated during the simulation according to the sensor models. It uses the LiDAR and RADAR sensor model blocks from the Simulink Sim3D Toolbox. Additionally, it records the position and attitude of the crane vessel by assuming the presence of an ideal MRU sensor. Note that creating a comprehensive model for the MRU sensor is beyond the scope of this thesis. Nonetheless, if a more precise model is needed, it can be readily replaced in a subsequent stage.

Lastly, the “Relative Heave Ground Truth” subsystem in the bottom right generates the actual values of relative heave distance (D) and speed ($\dot{D}$) within the simulation environment using the mathematical expressions derived in the preceding section 3.1. These ground truth values will serve as a reference for assessing the measurement methods in the subsequent stages. The `D_D_dot_Calculation` block takes the exact ship motions generated within Simulink as input and computes the exact D and $\dot{D}$.

3.2.3. Simulated Data Verification

The simulation workflow allows for generating LiDAR and RADAR data representing the Ship-to-Ship transfer of cargo scenario. This data acquisition can be made in a repeatable and reproducible manner, accommodating different ship configurations and dynamics. While the author did not develop the sensor models (MathWorks created them), ensuring that the generated data aligns with expectations is still essential. This verification is necessary to check if the simulation scenario is set up correctly for the development of data processing methods later on.

Various verification steps were conducted on the simulated data to ensure the accuracy and functionality of both the LiDAR and RADAR Point clouds. The specific verification tests for the LiDAR and RADAR, along with the anticipated outcomes, are summarised in Table 3.2 and Table 3.3, respectively.

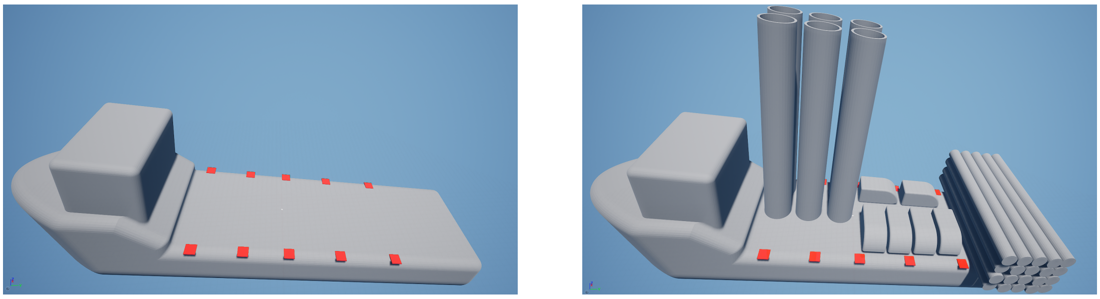
Table 3.2: LiDAR Verification Expectations

Test	Description	Expected Output
1	Point Cloud Shape	Accurately represent the shape and geometry of the supplier ship.
2	LiDAR Parameters	FOV: Full coverage of the ship deck space. Range: Capture objects within the specified range. Resolution: Capture fine details accurately.
3	Translation Motion	The point cloud accurately reflects the translation of the ship without any distortion or artefacts.
4	Rotation Motion	The point cloud accurately reflects the ship's orientation after rotation without distortion. Additionally, the centre of rotation remains at a constant distance from the LiDAR.
5	Objects Obstruction	When objects are on the ship deck, the LiDAR point cloud will display specific areas that are either partially or fully obscured by larger objects, depending on the viewing perspective.
6	Point Cloud Intensity	Reflective surfaces exhibit higher intensity, providing a realistic representation of reflectors.

Table 3.3: RADAR Verification Expectations

Test	Description	Expected Output
1	Point Cloud Shape	Accurately represent the shape and geometry of the supplier ship.
2	RADAR Parameters	FOV: Partial coverage of the ship deck. Range: Capture objects within the specified range. Resolution: Capture sparse point cloud.
3	Translation Motion	The point cloud accurately reflects the translation of the ship without any distortion or artefacts.
4	Rotation Motion	The point cloud accurately reflects the ship's orientation after rotation without distortion. Additionally, the centre of rotation remains at a constant distance from the RADAR.
5	Objects Obstruction	When objects are on the ship deck, the RADAR point cloud will display specific areas that are either partially or fully obscured by larger objects, depending on the viewing perspective.
6	Doppler Velocity	For each reflection (point), the projection of the body's velocity onto the ray orientation should match the measured Doppler velocity.

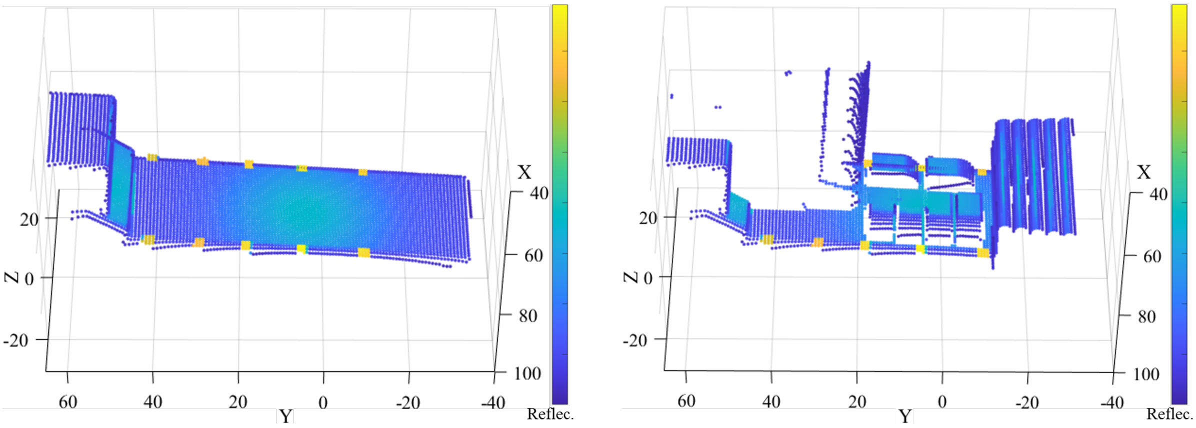
The verification tests used Simulink's default sensor parameters to simulate LiDAR and RADAR sensor data over a user-defined period. The sensor data was recorded in point cloud format for each time frame and subsequently used in the verification tests. Visual representations of the raw LiDAR and RADAR sensor data obtained at time step t are presented in Figure 3.9, Figure 3.10, and Figure 3.11.



(a)

(b)

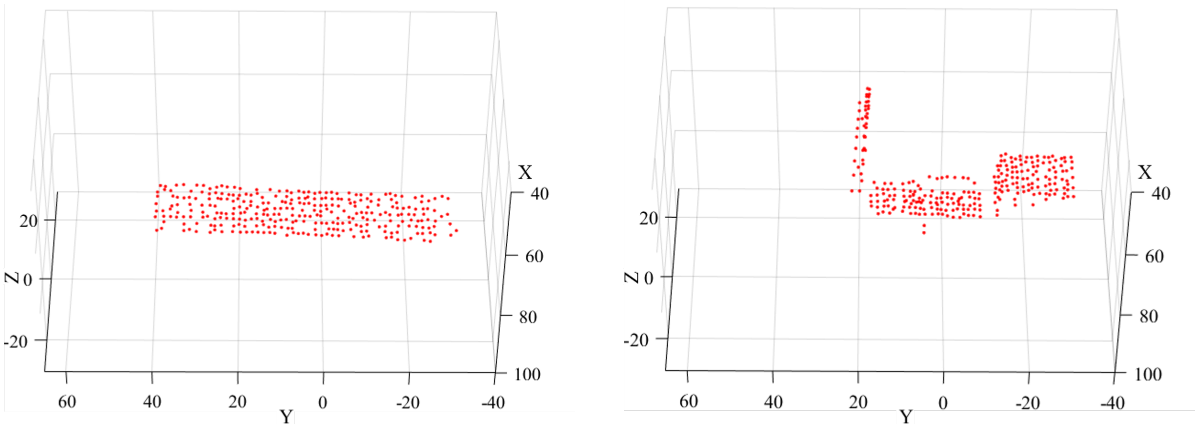
Figure 3.9: Ship model oblique view in the 3D environment of Unreal Engine at time step t. Two models are considered: (a) Ship with empty deck space; and (b) Ship with full deck space.



(a)

(b)

Figure 3.10: Generated LiDAR point cloud oblique view at time step t, where the colour scale indicates reflectivity. Two models are considered: (a) Ship with empty deck space; and (b) Ship with full deck space.



(a)

(b)

Figure 3.11: Generated RADAR point cloud oblique view at time step t. Two models are considered: (a) Ship with empty deck space; and (b) Ship with full deck space.

After conducting the extensive verification tests outlined in the previous tables, the outcomes for LiDAR and RADAR models are presented in Table 3.4 and Table 3.5, respectively. All tests were found to be in line with the predetermined expectations, except for test 6, concerning the RADAR sensor model. In order to identify the cause of the discrepancy in the measured Doppler velocities by the RADAR model, an extensive investigation was conducted. Discussions with experts from TUDelft and MathWorks led to the discovery of a bug in the RADAR Toolbox of MATLAB software and later its solution. Please consult Appendix D for a more comprehensive description.

Table 3.4: Simulated LiDAR Outputs

Test	Description	Simulated Output	Pass/Fail
1	Point Cloud Shape	Resembles the ship deck viewed from LiDAR position.	Pass
2	LiDAR Parameters	High-density point cloud providing accurate coverage of the ship deck.	Pass
3	Translation Motion	Point cloud accurately reflects each individual translation motion without distortion.	Pass
4	Rotation Motion	Point cloud accurately reflects each individual rotation motion without distortion. The center of rotation remains consistent.	Pass
5	Objects Obstruction	The presence of larger objects obstructed the LiDAR's line of sight, causing a shadow on the ship deck behind.	Pass
6	Point Cloud Intensity	Reflectors are clearly distinguishable in the point cloud intensity plot.	Pass

Table 3.5: Simulated RADAR Outputs

Test	Description	Simulated Output	Pass/Fail
1	Point Cloud Shape	Resembles the ship deck viewed from RADAR position.	Pass
2	RADAR Parameters	Low-density point cloud providing partial coverage of the ship deck.	Pass
3	Translation Motion	Point cloud accurately reflects each individual translation motion without distortion.	Pass
4	Rotation Motion	Point cloud accurately reflects each individual rotation motion without distortion. The center of rotation remains consistent.	Pass
5	Objects Obstruction	The presence of larger objects obstructed the RADAR's line of sight, causing a shadow on the ship deck behind.	Pass
6	Doppler Velocity	Linear velocities: Projection onto ray's orientation matches measured Doppler velocity. Angular velocities: Measured Doppler velocity does not correspond to angular velocity multiplied by the arm projected onto the ray's orientation.	Fail

3.2.4. Relative Heave Model Verification

The relative heave distance and speed equations were derived in the previous subsection 3.1.3, resulting in Equation 3.20 and Equation 3.24. These equations hold significant value for the current application, as all the terms on which the D and $\dot{D}$ equations depend are defined within the simulation and therefore known. Consequently, it becomes possible to determine D and $\dot{D}$ exact values within the simulation, enabling the evaluation of the developed measuring solutions. However, before proceeding to develop and assess solutions based on this model, it is crucial to verify the correctness of the output.

Various verification cases were conducted to ensure the model's accuracy and functionality. The specific verification procedures are summarised in Table 3.6. The verification tests were conducted by observing each DOF separately, using sinusoidal input waves for simplicity. The outcomes of the various tests, along with their compliance with expectations, are presented in Table 3.7.

Table 3.6: Model Verification Expectations

Test	Description	Expected Output
1	Crane Vessel: Static Supplier Vessel: Static	D constant and D = z_{p_t} + $(z_{b_1} - z_{b_2})$
2	Crane Vessel: Static Supplier Vessel: Sway/Surge/Heave	D has the same frequency and phase as the input wave. For heave, it has the same amplitude, shifted by a constant distance offset.
3	Crane Vessel: Sway/Surge/Heave Supplier Vessel: Static	D has the same frequency and phase as the input wave. For heave, it has the same amplitude, shifted by a constant distance offset.
4	Crane Vessel: Static Supplier Vessel: Roll/Pitch/Yaw	D has the same frequency, and phase as the input wave. However, it is shifted up by a constant initial distance offset and the amplitude will depend on the lever arm.
5	Crane Vessel: Roll/Pitch/Yaw Supplier Vessel: Static	D has the same frequency, and phase as the input wave. However, it is shifted up by a constant initial distance offset and the amplitude will depend on the lever arm.
6	Crane Vessel: Static Supplier Vessel: Roll/Pitch/Yaw	If the lever arm is zero then D is constant
7	Crane Vessel: Roll/Pitch/Yaw Supplier Vessel: Static	If the lever arm is zero then D is constant

Table 3.7: Model Verification Outputs

Test	Description	Output	Pass/Fail
1	Crane Vessel: Static Supplier Vessel: Static	D constant and D = z_{p_t} + $(z_{b_1} - z_{b_2})$, if Roll and Pitch is zero.	Pass
2	Crane Vessel: Static Supplier Vessel: Sway/Surge/Heave	D matches the respective input wave characteristics.	Pass
3	Crane Vessel: Sway/Surge/Heave Supplier Vessel: Static	D matches the respective input wave characteristics.	Pass
4	Crane Vessel: Static Supplier Vessel: Roll/Pitch/Yaw	D matches the frequency and phase of the input wave with the correct distance offset. The amplitude is smaller when the lever arm is smaller and vice-versa.	Pass
5	Crane Vessel: Roll/Pitch/Yaw Supplier Vessel: Static	D matches the frequency and phase of the input wave with the correct distance offset. The amplitude is smaller when the lever arm is smaller and vice-versa.	Pass
6	Crane Vessel: Static Supplier Vessel: Roll/Pitch/Yaw	The lever arm is zero, then D is constant. If Roll and Pitch is zero then D is constant regardless of yaw	Pass
7	Crane Vessel: Roll/Pitch/Yaw Supplier Vessel: Static	The lever arm is zero, then D is constant. If Roll and Pitch is zero then D is constant regardless of yaw	Pass

3.3. Discussion

This chapter establishes the foundation for accomplishing the main research goal, laying the groundwork for developing novel measuring solutions. Two critical prerequisites were addressed: the acquisition of LiDAR and RADAR sensor data, simulated in the case of this thesis, and the establishment of ground truth values essential for evaluating and comparing the developed solutions.

The first section of this chapter presents a kinematic model for the relative heave problem, addressing **Research Question 3**. To the best of the author's knowledge, this marks the first comprehensive relative heave model developed, providing insight into the various factors affecting relative heave. From this model, mathematical expressions for the relative heave distance (D) and speed ($\dot{D}$) are derived in terms of the position and attitude of each ship. These expressions are later integrated into the simulation environment to determine the exact relative heave values.

The second section introduces a modular simulation workflow utilising Simulink and Unreal Engine platforms to address **Research Question 4**. Notably, to the author, this simulation environment represents a novel implementation for this specific application. It enables the generation of RADAR and LiDAR data within the context of Ship-to-Ship cargo transfer scenarios, offering advantages in terms of repeatability, low cost, and flexibility compared to experimental setups. The study focuses on the analysis of two distinct ship and cargo combinations.

Finally, with representative sensor data and the exact relative heave values, the focus now shifts to developing and evaluating distinct measuring methods. The subsequent chapter 4 will concentrate on developing data processing and analysis algorithms for accurate relative heave estimation.

4

Relative Heave Measurement and Estimation

This chapter focuses on measuring and estimating the relative heave distance (D) and speed ($\dot{D}$) using the onboard crane vessel sensor data, addressing **Research Question 5.** Two sensor units are explored: RADAR-MRU and LiDAR-MRU. Since both sensor units do not provide direct measurements of D and $\dot{D}$, three novel measuring methods are presented in section 4.1: *deck detection*, *reflectors detection* and *Doppler-based* methods. These measuring methods concern the data processing and analysis stage to convert the acquired sensor data into the desired measurements D or $\dot{D}$. Lastly, section 4.2 presents the estimation stage, where a Kalman Filter (KF) is implemented to estimate both D and $\dot{D}$ using the measurements obtained from the implemented methods. Figure 4.1 depicts the measuring solutions structure, from data acquisition to the system states estimation, that will be discussed throughout this chapter.

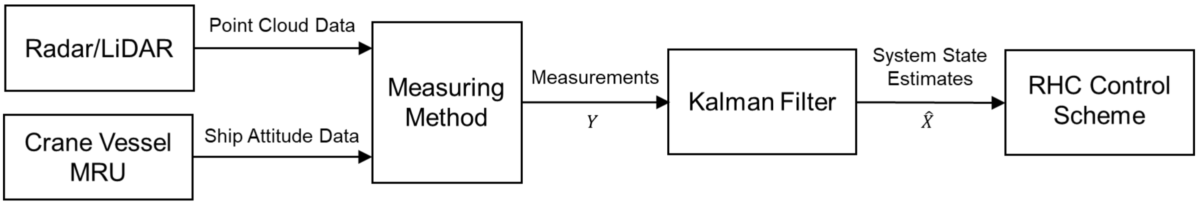


Figure 4.1: Measuring Solution Scheme depicting system measurements ($Y = D$ or $Y = \dot{D}$) and state estimates ($\hat{X} = [\hat{D} \;\; \hat{\dot{D}}]$) aimed at achieving accurate estimations of relative heave distance and speed.

4.1. Measuring Methods

The availability of MRU and point cloud data alone does not hold any intrinsic meaning. Thus, the data processing and analysis stage is vital in extracting valuable information from the raw sensor data. Various techniques are employed to achieve this goal, using the geometric understanding of the problem to establish significant relationships between the sensors' data and the desired system states.

Three measuring methods have been developed for this purpose, in particular *deck detection* (subsection 4.1.1), *reflectors detection* (subsection 4.1.2), and *Doppler-based* (subsection 4.1.3). The first method measures D and is implemented for both sensor units, relying on the position information of all detected points. The second method, exclusive to LiDAR-MRU, measures D by leveraging LiDAR's ability to detect differences in objects reflectivity. The last method measures $\dot{D}$ and is implemented solely for RADAR-MRU, as it can measure the Doppler velocity for each point in the point cloud.

4.1.1. Deck Detection Method

The line-plane intersection technique determines the intersection between a line with a specified direction and a plane. By determining the position of this intersection point, it becomes possible to compute its relative distance.

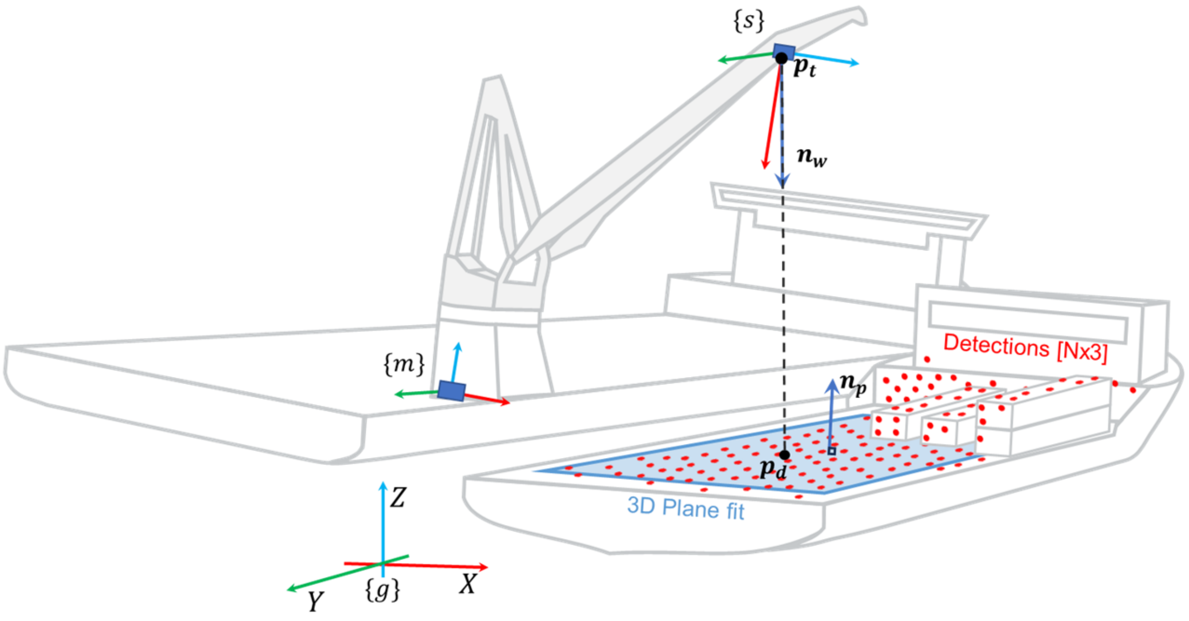


Figure 4.2: Overview of the *deck detection* method. The sensor captures N points (x, y, z) relative to its frame $\{s\}$, which are then used to define a best-fitting plane equation ($ax + by + cz + d = 0$), with $\boldsymbol{n_p}$ denoting the normal vector of the plane. The intersection point $\boldsymbol{p_d}$ is determined through the line-plane intersection of vector $\boldsymbol{n_w}$ and the fitted 3D plane. By knowing both $\boldsymbol{p_t}$ and $\boldsymbol{p_d}$, the value of D is computed.

The same geometric principle can be applied to determine the relative heave distance (D). As depicted in Figure 4.2, this method involves defining the ship deck plane equation ($ax + by + cz + d = 0$) and the correct orientation of the crane-hook connecting wire (n_w). Considering both factors, the problem of determining the relative heave distance can be effectively addressed.

When applying this method, a significant challenge lies in detecting the plane of the feeder vessel deck itself. However, both RADAR and LiDAR sensors excel at providing a detailed 3D graphical representation of the surrounding environment. By leveraging the positional information (XYZ) of all the measured points within the generated point cloud, it becomes possible to infer the position of the ship deck relative to the sensor frame.

Assuming the sensor mounted on the boom tip of the supplier vessel has a direct view of the supplier vessel, it is likely that a substantial portion of the data points captured by the sensor pertains to the deck of the feeder vessel and the objects being transported. An efficient method to differentiate the ship deck from the transported objects is to employ a plane fitting algorithm. This algorithm analyses all the captured points to identify the ideal plane model that aligns with the most significant plane in view. Since the ship deck is usually the most prominent flat surface in the scene, it is highly probable to be detected by this algorithm. Consequently, it effectively separates the ship deck points from those associated with the transported objects and provides the optimal plane equation relative to the sensor frame.

Fitting a plane model to a set of points is a well-researched case that finds applications in various fields [101]. One such application is in the autonomous driving industry, where accurately detecting the road plane is crucial for identifying objects in the surrounding environment. The RANdom SAmple Consensus (RANSAC) algorithm has emerged as a widely recognised and effective solution for these shape-fitting problems. Algorithm 2 in Appendix F presents a RANSAC Pseudocode for a plane model estimation in the presence of outliers and noise. This iterative method operates by randomly selecting subsets of data points and evaluating the number of inliers, which are data points consistent with the currently estimated plane model. The final output of the RANSAC algorithm is the plane model with the highest count of inliers obtained throughout the iterative process. A variant of the aforementioned algorithm is available in MATLAB called `pcfitplane`, which has been used to implement this measuring method.

Upon detecting a plane and obtaining its equation in the sensor frame $\{s\}$, the next step is determining the intersection with the orientation vector $\hat{\boldsymbol{n}}_w^s$ in the sensor frame. However, since the sensor frame is in motion, it is necessary to convert the vector $\hat{\boldsymbol{n}}_w^g = [0, 0, -1]^T$ into the appropriate coordinate system before applying the line-plane intersection. The conversion of $\hat{\boldsymbol{n}}_w$ between the global and crane vessel frames can be described as follows:

$$\hat{\boldsymbol{n}}_w^{b_1} = \boldsymbol{R}_g^{b_1} \hat{\boldsymbol{n}}_w^g \tag{4.1}$$

Where the rotation matrix $\boldsymbol{R}_g^{b_1}$ can be determined from Equation 3.16, using the Euler angles $(\phi_{b_1}, \theta_{b_1}, \psi_{b_1})$ measured by the MRU sensor. Furthermore, assuming the sensor is fixed to the crane with a known rotation $\boldsymbol{R}_{b_1}^s$ relative to the ship frame $\{b_1\}$:

$$\hat{\boldsymbol{n}}_w^s = \boldsymbol{R}_{b_1}^s \hat{\boldsymbol{n}}_w^{b1} \tag{4.2}$$

Combining Equations 4.1 and 4.2, the orientation vector $\hat{\boldsymbol{n}}_w^s$ can be described as follows:

$$\hat{\boldsymbol{n}}_w^s = \boldsymbol{R}_{b_1}^s \boldsymbol{R}_g^{b_1} \hat{\boldsymbol{n}}_w^g \tag{4.3}$$

With the plane equation and the orientation vector in the sensor frame $\{s\}$ established, the remaining task is to compute the distance between the sensor frame and the intersection point. Let $\boldsymbol{p_d}$ be the intersection point, which needs to lie on the line in the direction of $\hat{\boldsymbol{n}}_w$. Therefore:

$$\boldsymbol{p}_d^s = \boldsymbol{p}_t^s + D\hat{\boldsymbol{n}}_w^s \tag{4.4}$$

Assuming the crane tip point $\boldsymbol{p_t}$ is located at the origin of the sensor frame $\{s\}$, $\boldsymbol{p}_t^s = [0, 0, 0]^T$. And thus:

$$\boldsymbol{p}_d^s = D\hat{\boldsymbol{n}}_w^s \tag{4.5}$$

Furthermore, since $\boldsymbol{p}_d^s$ lies on the plane, it must also satisfy the plane equation:

$$\boldsymbol{n}_p^s \cdot \boldsymbol{p}_d^s + d = 0 \tag{4.6}$$

Here, $\boldsymbol{n}_p^s = [a, b, c]^T$ is the plane's normal vector, and a, b, c, d are the parameters of the plane equation ($ax + by + cz + d = 0$). Substituting Equation 4.5 into Equation 4.6:

$$D\boldsymbol{n}_p^s \cdot \hat{\boldsymbol{n}}_w^s + d = 0 \tag{4.7}$$

Solving for D and expanding the vector notation:

$$D = \frac{-d}{a\hat{n}_{w_x}^s + b\hat{n}_{w_y}^s + c\hat{n}_{w_z}^s} \tag{4.8}$$

Finally, the problem of measuring the relative heave distance (D) from a point cloud scene capture can be effectively solved by integrating the plane fitting algorithm with the line-plane intersection solution. The implementation steps for the LiDAR-MRU are outlined in Figure 4.3, while Figure 4.4 provides the steps for the RADAR-MRU. Despite their similarity, the distinct nature of the RADAR point cloud necessitates slight variations in the processing stage.

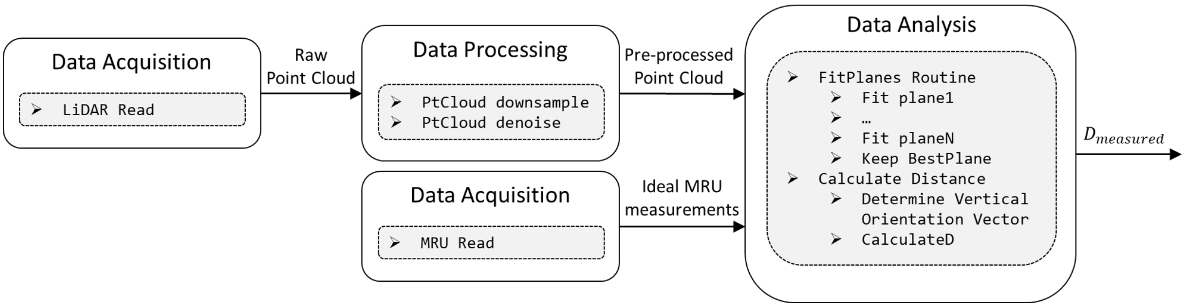


Figure 4.3: *Deck detection* implementation for the LiDAR-MRU sensor unit. The raw data is acquired from the sensors, processed and analysed to determine the relative heave distance (D).

The LiDAR-MRU implementation involves processing the acquired point cloud to achieve two main objectives: reducing its size for computational efficiency and enhancing data quality by eliminating noise and outliers while preserving underlying structures.

In order to optimise computational efficiency, a random elimination process is employed to reduce the point cloud size by a desired percentage. This procedure is implemented in MATLAB's `pcdownsample` built-in function. In addition, points are eliminated based on a threshold derived from the average distance to their k-nearest neighbours to remove noise from the point cloud. This algorithm is implemented within MATLAB's `pcdenoise` built-in function.

The analysis stage follows the processing stage and involves several vital steps. First, the point cloud data undergoes a multiple plane fitting routine employing the RANSAC algorithm (`pcfitplane`) N times on the same point cloud. This algorithm iterates N times to find the best plane representing the ship deck based on a specific metric (plane with the highest D). The goal is to capture the ship deck accurately while excluding any surfaces detected from the carried objects. Please refer to Appendix F Algorithm 3 for more detailed information.

Once the plane model is obtained, the final calculation of the relative heave distance (D) uses the line-plane intersection method. The implemented algorithm uses Equation 4.8 to determine the intersection point between the line and the fitted plane, computing its relative distance to the crane tip point.

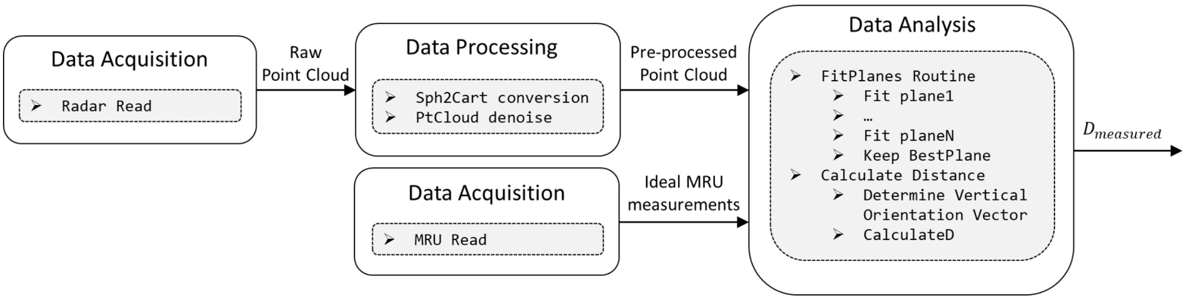


Figure 4.4: *Deck detection* implementation for the RADAR-MRU sensor unit. The raw data is acquired from the sensors, processed and analysed to determine the relative heave distance (D).

The implementation steps for the RADAR-MRU sensor unit closely resemble those for the LiDAR-MRU sensor unit, with a notable distinction in the processing stage. In contrast to the dense point cloud generated by LiDAR, the sparse nature of the RADAR point cloud renders the size reduction technique inapplicable. A conversion step is also incorporated to transform the data points from spherical coordinates to Cartesian coordinates, as the RADAR model produces data in this format. It is worth noting that MATLAB offers a built-in function called `Sph2Cart` for this purpose.

4.1.2. Reflectors Detection Method

The *reflectors detection* method leverages a distinctive aspect of LiDAR's working principle. LiDAR sensors continuously emit laser pulses to calculate the relative distance to an object using the time-of-flight principle. However, when these pulses are reflected, a certain amount of energy is inevitably lost. This loss enables LiDAR to detect changes in the intensity of the reflected signals. As a result, it becomes feasible to identify highly reflective objects by analysing the corresponding high-intensity points within the point cloud. Refer to Figure 4.5 for a visual depiction.

Expanding on this reflectivity concept, the proposal to utilise retro-reflectors on the ship deck emerged to detect the deck without relying on all captured points. Instead, only the points obtained from the retro-reflectors are taken into account. While these retro-reflectors must be installed or temporally placed on the supplier vessel, the intent of eliminating reliance on external sensors remains fulfilled. Furthermore, the developed method requires no specific pattern for setting these reflectors. Only an adequate quantity, spread evenly, to ensure continuous visibility from the LiDAR to establish a plane.

The developed measuring method in this section closely resembles the previous *deck detection* method. However, a key difference exists: the plane model is now obtained exclusively using points identified as reflectors/markers. This modification is expected to enhance both computational efficiency and robustness. After fitting the plane to the markers, the line-plane intersection method is again

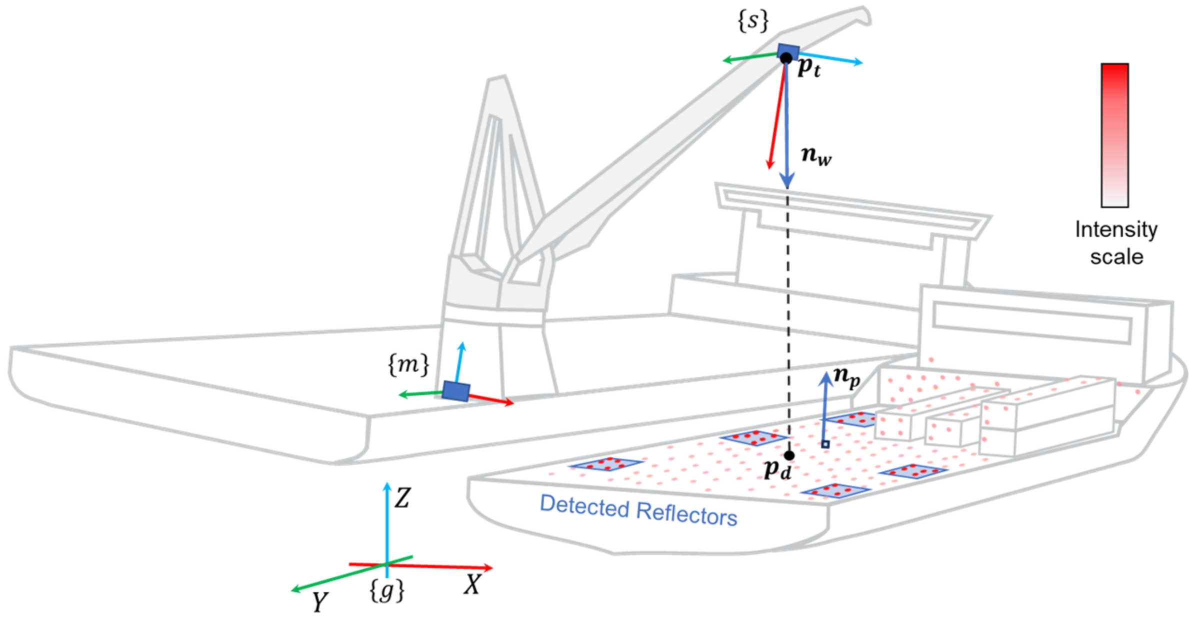


Figure 4.5: Overview of the *reflector detection* method. The sensor captures the reflectivity of all points, and the reflectors are detected relative to its frame $\{s\}$. The reflector points are then utilised to define the ship deck plane equation ($ax + by + cz + d = 0$), where $\boldsymbol{n_p}$ represents the normal vector of the plane. The intersection point $\boldsymbol{p_d}$ is determined through the line-plane intersection of vector $\boldsymbol{n_w}$ and the fitted plane of reflectors. By knowing both $\boldsymbol{p_t}$ and $\boldsymbol{p_d}$, the value of D is computed.

employed to compute the relative heave distance, denoted as D.

This method relies on intensity readings from the point cloud, so it has only been implemented with the LiDAR-MRU sensor unit. Figure 4.6 outlines the measuring method steps. The main objective is to detect the reflectors placed on the ship deck. To achieve this, Algorithm 4 shown in the Appendix F is employed to identify the highly-reflective points based on an intensity threshold. Subsequently, all other non-reflective points are filtered out. The data analysis follows the processing stage and involves two vital steps. Instead of fitting multiple planes as done in previous methods, the `pcfitplane` function from MATLAB is applied once. Once the most suitable plane model is obtained, the final calculation of the relative heave distance (D) uses the line-plane intersection algorithm described before.

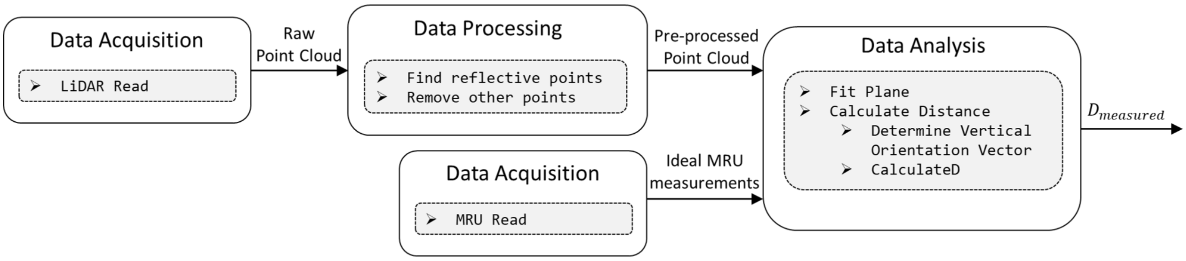


Figure 4.6: *Reflectors detection* implementation for the LiDAR-MRU sensor unit. The raw data is acquired from the sensors, processed and analysed to determine the relative heave distance (D).

4.1.3. Doppler-Based Method

The *Doppler-based* method utilises a particular feature of RADAR sensing technology. Like LiDARs, it uses the time-of-flight principle to determine the relative distance of an object. However, it can also detect the frequency shift between the electromagnetic wave that is emitted and received.

Measuring this frequency shift is a way to determine the relative velocity between the source and target objects. The change in frequency is a result of the Doppler effect, where a signal source moving towards or away from the target or vice versa can cause an apparent change in frequency. The following expression describes how the measured frequency shift depends on the relative velocity between the source and target:

$$\Delta f = \frac{v}{c} f_s \tag{4.9}$$

Where c is the propagation speed of waves in the medium, f_s is the source signal frequency,Δf is the frequency shift, and v is the object's velocity relative to the source. However, this equation only holds if the object is moving along the beam direction of the RADAR, which, as seen in Figure 4.7, is often not the case.

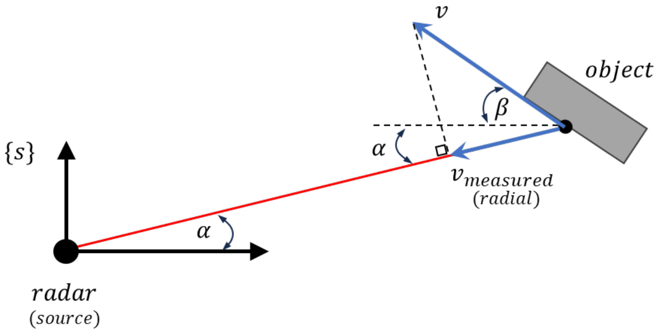


Figure 4.7: Representation of the measured Doppler velocity (v_{measured}) along the RADAR's beam direction (α) for an object moving with a relative velocity (v) and direction (β). The Doppler velocity is depicted as the projection of the relative velocity onto the beam direction.

This means the measured velocity by the RADAR ($v_{measured}$), deduced from Equation 4.9, is not equal to the actual relative velocity of the object v. Instead, it is the projection of the object's speed onto the RADAR beam direction. Taking Figure 4.7 as an example:

$$\Delta f = \frac{v}{c} f_s \cdot cos(\beta + \alpha) \quad \Rightarrow \quad v_{measured} = v \cdot cos(\beta + \alpha) \tag{4.10}$$

If only a single reflection of the object is captured by the RADAR, it is impossible to determine its actual velocity. However, the presence of multiple reflections enables the inference of the object's velocity by considering the radial orientation of each reflection. It is important to note that the number of reflections required to solve the equation depends on the degrees of freedom. Considering the previous example, assuming no rotational motion, the actual relative velocity could be determined with only two reflections, see Figure 4.8.

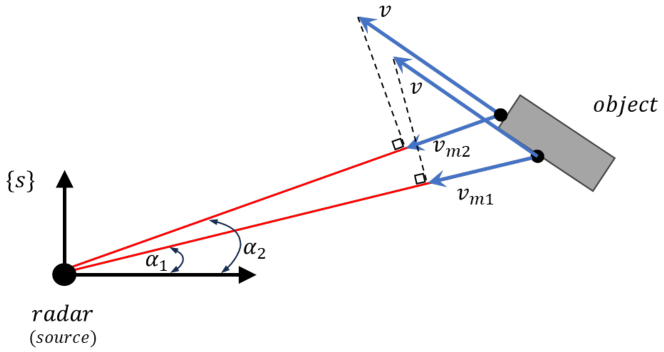


Figure 4.8: Representation of multiple RADAR detections on a moving object with a relative velocity (v) and direction (β). The measured Doppler velocity (v_m) is different for each detection. However, they are both the projection of the relative velocity onto the beam directions (α_1, α_2).

Figure 4.8 yields the following system of equations, consisting of two equations and two unknowns (v, β). As a result, the relative velocity can be determined.

$$\begin{cases} v_{m1} = v \cdot cos(\beta + \alpha_1) \\ v_{m2} = v \cdot cos(\beta + \alpha_2) \end{cases} \tag{4.11}$$

This same principle applied in the previous simplified example can also be applied to solve the relative heave problem and measure the relative heave speed ($\dot{D}$) of the supplier vessel.

Considering the feeder ship's substantial size, it is reasonable to assume that most reflected signals originate from its own body, as shown in Figure 4.9. This characteristic can be leveraged during the development of this method. Building upon the previous Doppler effect fundamental knowledge, the author comprehended the significance of establishing a relationship between the various Doppler velocity measurements (v_m) and the feeder vessel's actual relative linear and angular velocities.

Once the correlation function is determined and sufficient data points are collected, a solver can compute the ship's relative velocities that best match the RADAR measurements. Subsequently, the relative heave speed ($\dot{D}$) can be computed.

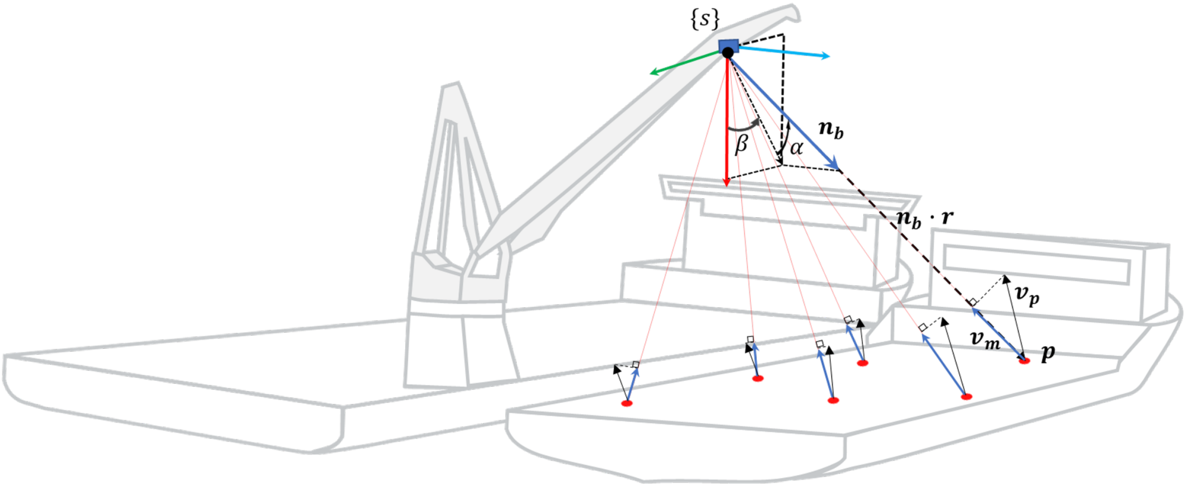


Figure 4.9: Overview of RADAR measurements in 3D. The red circles represent the sensor's detections. Each one provides the relative position of a point in spherical coordinates (α, β, r) with respect to its frame $\{s\}$. Additionally, the sensor captures the relative Doppler velocity $\boldsymbol{v}_m$ for each detected point, which is the projection of the point's velocity $\boldsymbol{v}_p$ onto the beam direction vector $\boldsymbol{n}_b$.

Figure 4.9 depicts the 3D RADAR measurements given in spherical coordinates. The azimuth (β) and elevation (α) angle of the reflected beam direction ($\boldsymbol{n}_b$) are captured along with the radial distance (r) to the object. Using a Cartesian left-handed coordinate system, the unit vector $\hat{\boldsymbol{n}}_b$ can be expressed in the sensor frame $\{s\}$ as:

$$\hat{\boldsymbol{n}}_b^s = \begin{bmatrix} cos(\alpha)cos(\beta) \\ -cos(\alpha)sin(\beta) \\ sin(\alpha) \end{bmatrix} \tag{4.12}$$

Combining Equation 4.12 and the measured radial distance (r), the position of a generic point $\boldsymbol{p}$ in the sensor frame $\{s\}$ can be given in Cartesian coordinates.

$$\boldsymbol{r}_{p/s}^s = \hat{\boldsymbol{n}}_b^s \cdot r = \begin{bmatrix} cos(\alpha)cos(\beta) \\ -cos(\alpha)sin(\beta) \\ sin(\alpha) \end{bmatrix} \cdot r \tag{4.13}$$

Additionally, the RADAR also captures the point's relative velocity ($\boldsymbol{v}_p$) projected onto the RADAR beam direction ($\boldsymbol{n}_b$). This measurement ($\boldsymbol{v}_m$) can be expressed mathematically as:

$$\boldsymbol{v}_m^s = \boldsymbol{v}_p^s \cdot \hat{\boldsymbol{n}}_b^s \tag{4.14}$$

Note: $\boldsymbol{v}_p^s = \dot{\boldsymbol{r}}_{p/s}^s$, where $\dot{\boldsymbol{r}}$ is a vector

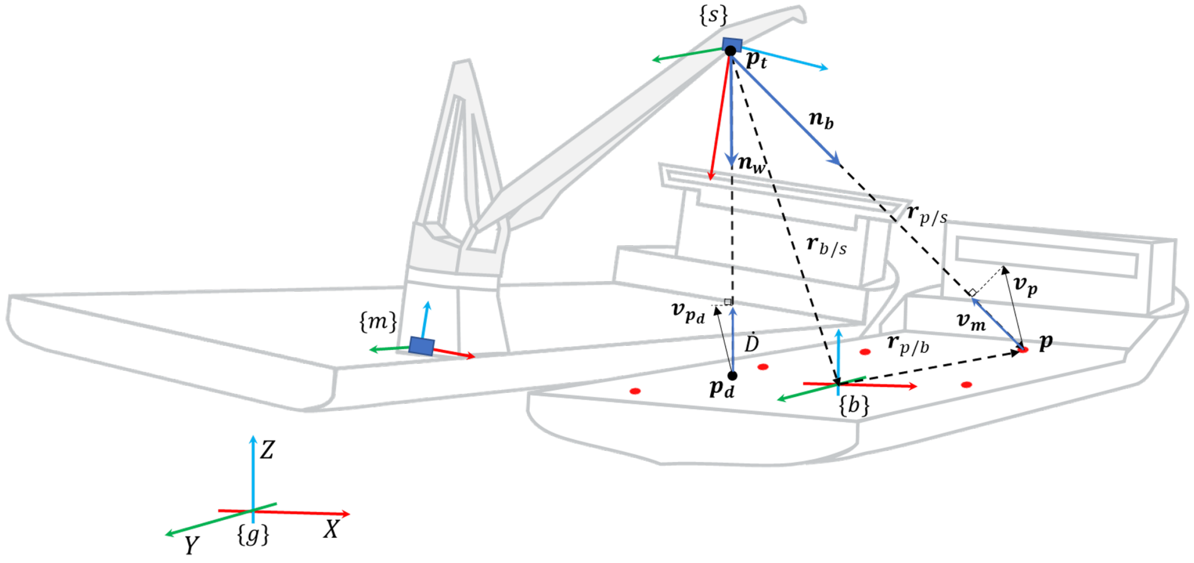


Figure 4.10: Overview of the *Doppler-based* method derivation. The relative ship motions in the body-fixed frame $\{b\}$ are correlated with RADAR measurements at point $\boldsymbol{p}$ in the direction of $\boldsymbol{n}_b$, enabling the determination of the ship's relative velocities. Subsequently, the relative velocity of the point directly below the RADAR ($\boldsymbol{v}_{p_d}$) is computed, and its projection onto $\boldsymbol{n_w}$ provides an approximation of the relative heave rate $\dot{D}$.

Considering the ship as a rigid body, each point ($\boldsymbol{p}$) on the body experiences the same linear velocity ($\dot{\boldsymbol{r}}^s_{b/s}$) and angular velocity ($\dot{\boldsymbol{\Theta}}_{sb}$) relative to the body-fixed frame (refer to Figure 4.10). Where the terms $\dot{\boldsymbol{r}}^s_{b/s}$ and $\dot{\boldsymbol{\Theta}}_{sb}$ represent the time derivatives of the body frame $\{b\}$ position and attitude, respectively, about the RADAR frame $\{s\}$. This assumption is particularly useful for deriving an expression that relates all measurements to the body motions.

$$\boldsymbol{r}^s_{b/s} = \begin{bmatrix} x_b \\ y_b \\ z_b \end{bmatrix}, \quad \boldsymbol{\Theta}_{sb} = \begin{bmatrix} \phi_b \\ \theta_b \\ \psi_b \end{bmatrix}, \quad \dot{\boldsymbol{r}}^s_{b/s} = \begin{bmatrix} \dot{x}_b \\ \dot{y}_b \\ \dot{z}_b \end{bmatrix}, \quad \boldsymbol{w}_{sb} = \begin{bmatrix} \dot{\phi}_b \\ \dot{\theta}_b \\ \dot{\psi}_b \end{bmatrix} \tag{4.15}$$

The goal is to establish a relationship between all the measured velocities ($\boldsymbol{v}_m$) and the relative body velocities. To accomplish this objective, the replacement of the variable $\dot{\boldsymbol{r}}^s_{p/s}$ in Equation 4.14 with $\dot{\boldsymbol{r}}^s_{b/s}$ and $\dot{\boldsymbol{\Theta}}_{sb}$ is proposed. Notably, the latter two parameters remain consistent for all the points under observation.

From Figure 4.10, the position of point $\boldsymbol{p}$ can be directly obtained from the RADAR measurement. Alternatively, it can be represented as the sum of two vectors: one describing the position of the body reference frame $\{b\}$ in the sensor frame $\{s\}$, and the other describing the position of point $\boldsymbol{p}$ in $\{b\}$.

$$\boldsymbol{r}^s_{p/s} = \boldsymbol{r}^s_{b/s} + \boldsymbol{R}^s_b \boldsymbol{r}^b_{p/b} \tag{4.16}$$

Since $\{b\}$ is fixed to the body, $\dot{\boldsymbol{r}}^b_{p/b} = 0$, the derivative of Equation 4.16 can be expressed as follows:

$$\dot{\boldsymbol{r}}^s_{p/s} = \dot{\boldsymbol{r}}^s_{b/s} + \dot{\boldsymbol{R}}^s_b \boldsymbol{r}^b_{p/b} \tag{4.17}$$

Please refer to the expanded system of equations presented in Appendix E, Equation E.1.

The velocity vector $\dot{\boldsymbol{r}}^s_{p/s}$ depends on the position vector $\boldsymbol{r}^b_{p/b}$ for each measured point. Therefore, it is essential to substitute it with a measurable quantity from the RADAR system. $\boldsymbol{r}^b_{p/b}$ can be determined by combining equations Equation 4.13 and Equation 4.16.

$$\boldsymbol{r}^s_{b/s} + \boldsymbol{R}^s_b \boldsymbol{r}^b_{p/b} = \begin{bmatrix} cos(\alpha)cos(\beta)r \\ -cos(\alpha)sin(\beta)r \\ sin(\alpha)r \end{bmatrix} \tag{4.18}$$

Please refer to the expanded system of equations presented in Appendix E, Equation E.2.

Solving Equation 4.18 yields the expression for $\boldsymbol{r}^b_{p/b}$ as a function of RADAR measurements and the body-fixed frame's position ($\boldsymbol{r}^b_{b/s}$) and attitude ($\boldsymbol{\Theta}_{sb}$) relative to the sensor frame.

$$\boldsymbol{r}^b_{p/b} = \boldsymbol{R}^{s-1}_b \begin{bmatrix} cos(\alpha)cos(\beta)r - x_b \\ -cos(\alpha)sin(\beta)r - y_b \\ sin(\alpha)r - z_b \end{bmatrix} \tag{4.19}$$

Please refer to the detailed solution presented in Appendix E, Equation E.3, considering its extended size.

After substituting the term $\boldsymbol{r}^b_{p/b}$ with the solution obtained from the previous Equation 4.19 in Equation 4.17, the resulting expression for $\dot{\boldsymbol{r}}^b_{p/b}$ depends solely on the body-fixed frame states. This expression can then be integrated into Equation 4.14.

The resulting expression establishes a clear connection between the measured velocity ($\boldsymbol{v}_m$) detected by the RADAR and the known orientation of the reflected beam (α, β), along with the relative position, attitude, linear, and angular velocities of the ship with respect to the RADAR frame.

$$\boldsymbol{v}_m = f(\dot{\boldsymbol{r}}^s_{b/s}, \boldsymbol{r}^s_{b/s}, \dot{\boldsymbol{\Theta}}_{sb}, \boldsymbol{\Theta}_{sb}, \alpha, \beta) \tag{4.20}$$

Where the complete expression can be written as:

$$\begin{aligned} v_m = {} & \dot{x}_b c_\beta c_\alpha - \dot{y}_b c_\alpha s_\beta + \dot{z}_b s_\alpha \\ & + \dot{\phi}_b (x_b (c_{\theta_b} s_\alpha s_{\psi_b} - c_\alpha s_\beta s_{\theta_b}) - y_b (c_\beta c_\alpha s_{\theta_b} + c_{\psi_b} c_{\theta_b} s_\alpha) - z_b (c_\beta c_\alpha c_{\theta_b} s_{\psi_b} + c_\alpha c_{\psi_b} s_\beta c_{\theta_b})) \\ & + \dot{\theta}_b (x_b c_{\psi_b} s_\alpha + y_b s_\alpha s_{\psi_b} + z_b (c_\alpha s_\beta s_{\psi_b} - c_\beta c_\alpha c_{\psi_b})) \\ & + \dot{\psi}_b (x_b c_\alpha s_\beta + y_b c_\beta c_\alpha) \end{aligned} \tag{4.21}$$

Considering the previous equation, the RADAR outputs for each point in the point cloud are denoted as v_m, α, and β. This results in a total of twelve unknowns, $\dot{x}_b, \dot{y}_b, \dot{z}_b, x_b, y_b, z_b, \dot{\phi}_b, \dot{\theta}_b, \dot{\psi}_b, \phi_b, \theta_b, \psi_b$. Since there are multiple points available, a solvable system can be formed by having a minimum of 12 points:

$$\underbrace{\begin{bmatrix} f(\dot{\boldsymbol{r}}^s_{b/s}, \boldsymbol{r}^s_{b/s}, \dot{\boldsymbol{\Theta}}_{sb}, \boldsymbol{\Theta}_{sb}, \alpha_1, \beta_1) \\ f(\dot{\boldsymbol{r}}^s_{b/s}, \boldsymbol{r}^s_{b/s}, \dot{\boldsymbol{\Theta}}_{sb}, \boldsymbol{\Theta}_{sb}, \alpha_2, \beta_2) \\ f(\dot{\boldsymbol{r}}^s_{b/s}, \boldsymbol{r}^s_{b/s}, \dot{\boldsymbol{\Theta}}_{sb}, \boldsymbol{\Theta}_{sb}, \alpha_3, \beta_3) \\ \vdots \\ f(\dot{\boldsymbol{r}}^s_{b/s}, \boldsymbol{r}^s_{b/s}, \dot{\boldsymbol{\Theta}}_{sb}, \boldsymbol{\Theta}_{sb}, \alpha_{12}, \beta_{12}) \end{bmatrix}}_{A} = \begin{bmatrix} v_{m1} \\ v_{m2} \\ v_{m3} \\ \vdots \\ v_{m12} \end{bmatrix} \tag{4.22}$$

One crucial question remains: for the system to be solvable, the aforementioned equations must be linearly independent. When one equation becomes a linear combination of others, the system loses its ability to be solved exactly. To verify this, the determinant of the matrix A should be non-zero. After computing the determinant and rank of matrix A, the results are as follows:

$$\text{det}(A) = 0 \quad \text{and} \quad \text{rank}(A) = 3 \tag{4.23}$$

Despite having sufficient RADAR measurements, determining the desired variables of the system remains difficult due to the limited linear independence in the previous set of equations, with only three out of twelve being independent. Nevertheless, certain assumptions can be made to make the system solvable, particularly by neglecting certain motions. After conducting numerous scenario tests, it became clear that the system is solvable only when assuming that the ship rotations are negligible compared to the translational motions of the ship, therefore assuming all points in the body experience the same velocity. Introducing rotational motions would require tracking the body-fixed reference, which is not feasible with Doppler velocity measurements alone.

Under the assumption of $\dot{\boldsymbol{\Theta}}_b = [0, 0, 0]^T$, the expression in Equation 4.21 can be rewritten as:

$$v_m = \dot{x}_b cos(\alpha)cos(\beta) - \dot{y}_b cos(\alpha)sin(\beta) + \dot{z}_b sin(\alpha) \tag{4.24}$$

This simplified expression results in a system of equations with three unknowns, which requires a minimum of three data points to be solvable.

$$\underbrace{\begin{bmatrix} cos(\alpha_1)cos(\beta_1) & -cos(\alpha_1)sin(\beta_1) & sin(\alpha_1) \\ cos(\alpha_2)cos(\beta_2) & -cos(\alpha_2)sin(\beta_2) & sin(\alpha_2) \\ cos(\alpha_3)cos(\beta_3) & -cos(\alpha_2)sin(\beta_3) & sin(\alpha_3) \end{bmatrix}}_{B} \begin{bmatrix} \dot{x}_b \\ \dot{y}_b \\ \dot{z}_b \end{bmatrix} = \begin{bmatrix} v_{m1} \\ v_{m2} \\ v_{m3} \end{bmatrix} \tag{4.25}$$

After computing the determinant and rank of matrix B, the results are as follows:

$$\text{det}(B) \neq 0 \quad \text{and} \quad \text{rank}(B) = 3 \tag{4.26}$$

After confirming the system's solvability, solving an overdetermined system of equations (a system with more equations than unknowns) involves employing methods that yield the most accurate approximate solution. A commonly used technique for this purpose is the "least squares method" [102]. Consider the following system of equations for n collected data points:

$$\underbrace{\begin{bmatrix} cos(\alpha_1)cos(\beta_1) & -cos(\alpha_1)sin(\beta_1) & sin(\alpha_1) \\ cos(\alpha_2)cos(\beta_2) & -cos(\alpha_2)sin(\beta_2) & sin(\alpha_2) \\ cos(\alpha_3)cos(\beta_3) & -cos(\alpha_2)sin(\beta_3) & sin(\alpha_3) \\ \vdots & \vdots & \vdots \\ cos(\alpha_n)cos(\beta_n) & -cos(\alpha_n)sin(\beta_n) & sin(\alpha_n) \end{bmatrix}}_{C} \begin{bmatrix} \dot{x}_b \\ \dot{y}_b \\ \dot{z}_b \end{bmatrix} = \underbrace{\begin{bmatrix} v_{m1} \\ v_{m2} \\ v_{m3} \\ \vdots \\ v_{mn} \end{bmatrix}}_{V_m} \tag{4.27}$$

The least squares solution for the previous system can be obtained by utilising the pseudo-inverse of matrix C, denoted as C^+. This method involves multiplying the pseudo-inverse (C^+) and the constant vector V_m. The resulting product yields the desired least squares solution.

$$\dot{\boldsymbol{r}}^s_{b/s} = C^+ V^s_m \tag{4.28}$$

The last step to obtain the approximate relative heave velocity is to project the relative velocity vector of the ship $\dot{\boldsymbol{r}}^s_{b/s}$ in the heave direction.

$$\dot{D} = \dot{\boldsymbol{r}}^s_{b/s} \cdot \boldsymbol{n}^s_w \tag{4.29}$$

The vector n^s_w used remains consistent with the previous methods, see Equation 4.3.

The *Doppler-based* method implementation steps for the RADAR-MRU sensor unit are outlined in Figure 4.11.

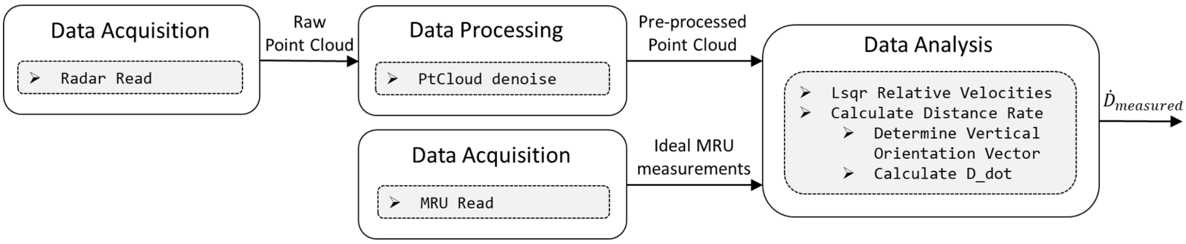


Figure 4.11: *Doppler-based* implementation for the RADAR-MRU sensor unit. The raw data is acquired from the sensors, processed and analysed to determine the relative heave speed ($\dot{D}$).

The implementation steps of this method differ significantly from previous approaches. However, the processing stage resembles the earlier RADAR-MRU implementation, with one notable difference: the exclusion of the point cloud's conversion to Cartesian coordinates. During the processing stage, noise elimination is performed, resulting in a noise-free point cloud where each point is represented in spherical coordinates, accompanied by the corresponding measured Doppler velocity (α, β, r, v_m).

The processed point cloud undergoes a subsequent analysis stage to determine the relative heave speed ($\dot{D}$). The ship's relative velocities are computed using the "Least Squares Method," as presented in Equation 4.28. Once the ship's relative velocities are obtained, the algorithm computes the relative heave speed by projecting these velocities onto the global frame vertical direction, as described in Equation 4.29.

4.2. Relative Heave Estimation

Using measurements directly as inputs for the control system represents a potentially practical and straightforward approach, mainly when such measurements are readily accessible. In the specific application under consideration, achieving real-time acquisition of relative heave distance and speed is an essential prerequisite for achieving a high level of control accuracy.

However, it is essential to note that the developed *deck detection* and *reflectors detection* methods yield measurements exclusively related to the relative heave distance. In contrast, the *Doppler-based* method focuses solely on determining the relative speed. As a result, whether one intends to independently implement these methodologies or integrate them at a later stage, leveraging estimation algorithms emerges as a valuable approach.

Moreover, in cases where measurements encounter sporadic reliability challenges—such as when the ship deck or reflectors go undetected temporarily—the estimation algorithm remains capable of producing short-term predictions. This capability, in turn, guarantees a seamless and continuous control process, thereby emphasising the value of implementing the estimation algorithm in the specific scenario.

4.2.1. Process Model

The real-time estimation of relative heave distance and speed is crucial for successful Ship-to-Ship cargo transfer. Commonly used estimation algorithms include the Kalman Filter (KF) and the Extended Kalman Filter (EKF) [103], which require both process and measurement models. These models relate the obtained measurements to a common vector, denoted as the state vector, that the algorithm aims to estimate. The continuous-time state vector used to describe the estimation problem is as follows:

$$\boldsymbol{x_s}(t) = \begin{bmatrix} D(t) \\ \dot{D}(t) \\ \ddot{D}(t) \end{bmatrix} \tag{4.30}$$

Here, D represents the relative heave distance, $\dot{D}$ stands for its speed, and $\ddot{D}$ indicates its acceleration. Although incorporating $\ddot{D}$ is not mandatory for the control system, it possesses the potential to serve as a valuable parameter within the assumed process model.

With the state vector established, deriving an appropriate process model remains necessary. Given the considerable size of both ships, an assumption is made regarding relatively constant relative heave acceleration, particularly over short time intervals. This assumption permits the formulation of the following continuous-time equations of motion:

$$\begin{cases} D(t) = D_0 + \dot{D}_0 t + \frac{1}{2}\ddot{D}t^2 \\ \dot{D}(t) = \dot{D}_0 + \ddot{D}t \\ \ddot{D}(t) = a \quad , \ a = const \end{cases} \tag{4.31}$$

By employing Euler integration, it becomes possible to discretise Equation 4.31. The underlying concept involves approximating the derivatives in continuous-time with their discrete-time counterparts:

$$\begin{cases} D[k+1] = D[k] + \dot{D}[k]T_s + \frac{1}{2}\ddot{D}[k]T_s^2 \\ \dot{D}[k+1] = \dot{D}[k] + \ddot{D}[k]T_s & , \ T_s = t_{k+1} - t_k \\ \ddot{D}[k+1] = \ddot{D}[k] \end{cases} \tag{4.32}$$

The state vector, defined in Equation 4.30, can also be reformulated for discrete-time as:

$$\boldsymbol{x_s}[k] = \begin{bmatrix} D[k] \\ \dot{D}[k] \\ \ddot{D}[k] \end{bmatrix} \tag{4.33}$$

Combining Equation 4.32 and Equation 4.33, the discrete-time state space model can be written as follows:

$$\boldsymbol{x_s}[k+1] = A\boldsymbol{x_s}[k] + G\boldsymbol{w} \tag{4.34}$$

$$A = \begin{bmatrix} 1 & T_s & \frac{1}{2}(T_s)^2 \\ 0 & 1 & T_s \\ 0 & 0 & 1 \end{bmatrix}, \quad G = \begin{bmatrix} \frac{1}{2}(T_s)^2 \\ T_s \\ 1 \end{bmatrix} \tag{4.35}$$

Where:

- $\boldsymbol{x_s}[k]$ is the state vector at time step k
- $\boldsymbol{x_s}[k+1]$ is the state vector at time step $k+1$
- A is the state transition matrix
- G is the process noise gain matrix
- $\boldsymbol{w}$ is the process noise

Process noise ($\boldsymbol{w}$) in this context represents a disturbance affecting the acceleration. This interpretation is accurate, as the acceleration naturally fluctuates during vessel movement.

4.2.2. Measurements Model

Following the development of the discrete-time process model, the next step in implementing the estimation algorithm involves deriving the measurement model. In this particular application, the measuring solutions rely on a combination of two sensors: MRU-LiDAR or MRU-RADAR. Consequently, the measurement model should incorporate these individual sensor measurements to obtain the desired system state estimates.

The measurement model is responsible for converting the relative heave distance or speed from the exteroceptive sensor frame to the global frame using the MRU pitch and roll measurements. It is important to note that, within the scope of this application, the MRU is assumed to be ideal and free from noise. This assumption simplifies the measurement model, enabling the direct acquisition of D or $\dot{D}$ while accounting for a single source of measurement noise.

Considering the aforementioned, the measurement model in discrete time can be written as follows:

$$y(k) = H\boldsymbol{x}(k) + v \tag{4.36}$$

For the *deck detection* and *reflectors detection* methods:

$$H = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \tag{4.37}$$

For the *Doppler-based* method:

$$H = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix} \tag{4.38}$$

Where:

- $y(k)$ is the measurement vector at time step k
- H is the measurement matrix
- v is the measurement noise

From Equation 4.38, only speed measurements are available. Therefore, the observability of distance states is compromised for the *Doppler-based* method. This occurs because the equation of motion involves both speed and initial position parameters, and without the latter, it becomes impossible to uniquely determine the distance.

While the integration of various techniques is beyond the scope of this thesis, it is essential to note that such implementation is feasible and could potentially solve the problem mentioned above. For example, to combine the *deck detection* and *Doppler-based* methods, the sole modification required in the estimation algorithm pertains to the measurement model, as follows:

$$H = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \tag{4.39}$$

4.2.3. Kalman Filter Implementation

Given the linear nature of the derived model, this section focuses on the practical implementation of the Kalman Filter algorithm for state estimation. The approach employs the previously derived time-discrete process and measurement models.

The algorithm below outlines the sequential procedures involved in state estimation. First, a prediction step is applied using the process model. Second, an update step is applied, where the initial predictions are corrected using real-time measurements obtained from the measurement model. If the measurements are not available, no correction is performed.

Algorithm 1 Linear Kalman Filter

```
Initialise:
$k \leftarrow 1$
$\hat{\boldsymbol{x}}_{k-1|k-1} \leftarrow \boldsymbol{x}_0$
$\boldsymbol{P}_{k-1|k-1} \leftarrow \boldsymbol{P}_0$
while not done do
    // Predict state and state covariance
    $\hat{\boldsymbol{x}}_{k|k-1} \leftarrow \boldsymbol{A}_k \hat{\boldsymbol{x}}_{k-1|k-1}$
    $\boldsymbol{P}_{k|k-1} \leftarrow \boldsymbol{A}_k \boldsymbol{P}_{k-1|k-1} \boldsymbol{A}_k^T + \boldsymbol{Q}_k$
    if measurement is valid then
        // Innovation covariance and Kalman gain
        $\boldsymbol{S}_k \leftarrow \boldsymbol{H}_k \boldsymbol{P}_{k|k-1} \boldsymbol{H}_k^T + \boldsymbol{R}_k$
        $\boldsymbol{K}_k \leftarrow \boldsymbol{P}_{k|k-1} \boldsymbol{H}_k^T \boldsymbol{S}_k^{-1}$
        // Update state estimate and state covariance
        $\boldsymbol{y}_k \leftarrow \boldsymbol{z}_k - \boldsymbol{H}_k \hat{\boldsymbol{x}}_{k|k-1}$
        $\hat{\boldsymbol{x}}_{k|k} \leftarrow \hat{\boldsymbol{x}}_{k|k-1} + \boldsymbol{K}_k \boldsymbol{y}_k$
        $\boldsymbol{P}_{k|k} \leftarrow (\boldsymbol{I} - \boldsymbol{K}_k \boldsymbol{H}_k) \boldsymbol{P}_{k|k-1}$
    else
        $\hat{\boldsymbol{x}}_{k|k} \leftarrow \hat{\boldsymbol{x}}_{k|k-1}$
        $\boldsymbol{P}_{k|k} \leftarrow \boldsymbol{P}_{k|k-1}$
    end if
    $k \leftarrow k + 1$
end while
```

4.3. Discussion

This chapter presents three distinct methods to measure relative heave, guiding the reader from the formulation of novel concepts to their practical implementation. Additionally, it outlines the implementation of a Kalman Filter used to estimate both the relative heave distance (D) and speed ($\dot{D}$). The developed measuring solutions align with the initial goal of relying solely on sensors aboard the crane vessel: an onboard Motion Reference Unit and an exteroceptive LiDAR or RADAR sensor.

In section 4.1, the three measuring methods are developed and implemented. While all three methods use point cloud data obtained from LiDAR or RADAR, along with MRU measurements, the approaches for computing pertinent system states differ. The first method, detailed in subsection 4.1.1, relies on the spatial coordinates of all captured points. This technique detects the ship deck plane and determines the relative heave distance (D). Importantly, it applies to both RADAR-MRU and LiDAR-MRU sensor units. The second method, discussed in subsection 4.1.2, is tailored for the LiDAR-MRU unit. It leverages the LiDAR's reflectivity information associated with each point, focusing on fitting a plane solely to points with high reflectivity. These high-reflectivity points correspond to reflections from reflectors or markers on the supplier's ship deck. Similarly to the first method, the relative heave distance (D) is computed. The third method, outlined in subsection 4.1.3, exclusively utilises the Doppler velocity measurements from the RADAR sensor. Its primary objective is to convert the point's measured Doppler velocities into the supplier vessel's relative velocity and compute the relative heave speed ($\dot{D}$).

Given the independent implementation of these measuring methods and the necessity of both D and $\dot{D}$ for real-time control, in section 4.2, a Kalman Filter estimation algorithm is implemented. This algorithm addresses the requirements above across all methods and holds significant practical value due to the potential unreliability and inaccuracy of measurements over short periods. In such instances, the estimation algorithm can provide predictions that enable smooth wire rope control.

In summary, this chapter introduces three novel measuring methods and an estimation algorithm designed to estimate relative heave distance and speed during Ship-to-Ship cargo transfers, effectively addressing **Research Question 5**. These measuring solutions transform raw data, initially lacking direct interpretability of the system states, into estimates of the desired system states. The subsequent chapter will verify the implemented measuring solutions and assess their processing speed and accuracy.

5 Measuring Solutions Verification and Evaluation

This chapter focuses on verifying and evaluating the measuring solutions introduced and implemented in the preceding chapter 4, summarised in Table 5.1. Firstly, section 5.1 describes the simulation setup, the selected Key Performance Indicators (KPIs), and the hardware specifications employed during algorithm execution. Secondly, section 5.2 presents the verification results for each measuring solution under ideal conditions, ensuring alignment with the expected outcomes. Subsequently, section 5.3 conducts a performance analysis, addressing **Research Question 6**, to assess the accuracy and computational efficiency of the implemented solutions. Furthermore, a sensitivity study complements this analysis, allowing for an understanding of the impact of various parameters on the measuring solutions' performance, addressing **Research Question 7**.

Table 5.1: Implemented measuring solutions overview

Measuring Solution Description	Sensor Unit	Measuring Method	Estimation Method
1. Deck Detection (LiDAR-MRU)	LiDAR-MRU	Deck Detection	Kalman Filter
2. Deck Detection (RADAR-MRU)	RADAR-MRU	Deck Detection	Kalman Filter
3. Reflectors Detection (LiDAR-MRU)	LiDAR-MRU	Reflectors Detection	Kalman Filter
4. Doppler-Based (RADAR-MRU)	RADAR-MRU	Doppler-Based	Kalman Filter

5.1. Verification and Evaluation Setup

This section provides essential preliminary information for the verification and evaluation cases of the implemented measuring solutions. It clarifies the Ship-to-Ship scenario and the sensor parameters used for each case. Furthermore, it introduces and justifies the selected KPIs. Lastly, since the computational times for each solution are examined, the hardware specifications operated during algorithm execution are also described.

5.1.1. Simulation Parameters

The primary goal of the measuring solutions' verification is to assure the accuracy of the algorithm's implementation. This validation is conducted through a simplified and idealised scenario purposefully designed to isolate potential sources of error. Ideal sensor models with noise-free measurements are employed, and a simplified scenario involving an object-free feeder vessel deck (depicted in Figure 3.5) is selected to minimise potential errors unrelated to the measuring method itself. Furthermore, the estimation stage is excluded in the verification case, focusing solely on verifying the obtained measurements.

Conversely, the measuring solutions' evaluation focuses on assessing their performance in real-world conditions. To this end, the selected sensor parameters are based on available sensors in the market (listed in Table 2.2). The exact parameters used can be found in Table 5.2 and Table 5.3. These sensor models account for factors such as noise and suboptimal resolution. Additionally, a loaded feeder vessel deck (depicted in Figure 3.6) is introduced to simulate the typical presence of components for wind turbine installation, thereby increasing the complexity of the fitting algorithms. Notably, the estimation step is included to ensure an equitable comparison of the measuring solutions.

Table 5.2: 3D Probabilistic RADAR Block Parameters

Parameter	Value
Angular resolution azimuth	1 [deg]
Angular resolution elevation	1.5 [deg]
Range resolution	0.5 [m]
Range rate resolution	0.17 [m/s]
Field of View (FOV)	60x12 [deg]
Detection Range	300 [m]
Detection Probability	0.9
Detection Probability range	250[m]

Table 5.3: 3D LiDAR Block Parameters

Parameter	Value
Angular resolution horizontal	0.5 [deg]
Angular resolution vertical	0.5 [deg]
Range resolution	0.02 [m]
Field of View (FOV)	120x30 [deg]
Detection Range	300 [m]
Reflectivity out-port	On

The ship motions are a significant parameter within the simulation setup and can influence the verification and evaluation results. It is worth noting that modelling the ship's dynamics falls outside the scope of this study. Instead, regular sinusoidal motions, validated in consultation with Huisman experts, are applied to all six degrees of freedom (6 DoF) with appropriate amplitudes and periods.

For the supplier vessel, these motions consist of a 2-meter amplitude for heave, 2.5 degrees for pitch, and 1 degree for roll, each with a period of approximately 10 seconds. In the case of the crane vessel, the motion characteristics are similar but phased 45 degrees apart from those of the supplier vessel, with slightly lower amplitudes. Please refer to Appendix G for a complete description of these motions.

The resulting ground truth values for relative heave distance and speed, obtained through the exact kinematic model developed in section 3.1, are as follows:

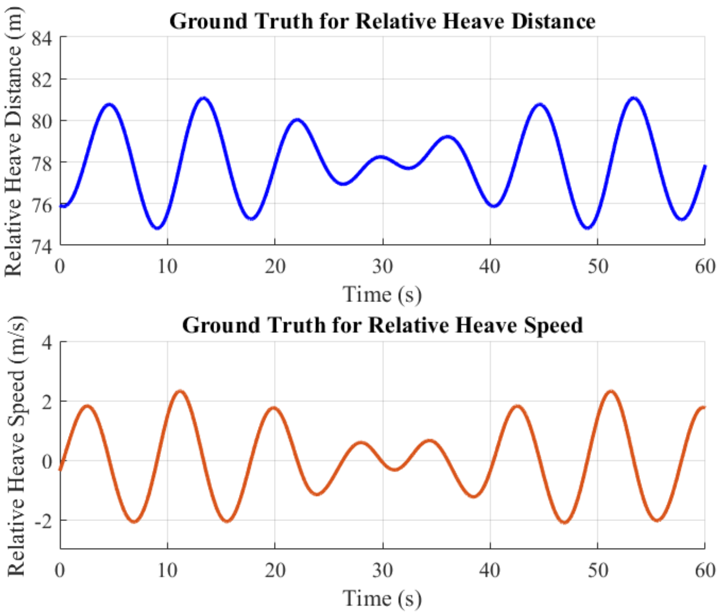


Figure 5.1: Illustration of the relative heave distance (D) and speed ($\dot{D}$) reference employed for the verification and evaluation of the implemented measuring solutions.

5.1.2. Key Performance Indicators (KPIs)

The selection of appropriate verification and evaluation metrics for the implemented solutions is essential to draw accurate conclusions in each case. In the verification phase, the primary focus is on assessing the accuracy of the solutions, aiming for zero measurement error in the ideal scenario. Similarly, in the evaluation stage, the goal is to compare the accuracy across different solutions, as well as their processing speed. Consequently, two KPIs have been selected to assess accuracy: Mean Absolute Error (MAE) and Maximum Absolute Error (Max Error). Furthermore, an extra KPI, Average Processing Time (APT), is exclusively employed for the evaluation phase.

Mean Absolute Error (MAE)

The MAE metric offers an understanding of the average discrepancy between measured or estimated values ($\hat{Y}$) and the actual values (Y). It is computed as follows:

$$\text{MAE} = \frac{1}{N}\sum_{i=1}^{N} |\hat{Y}_i - Y_i| \tag{5.1}$$

Maximum Absolute Error (Max Error)

The Maximum Absolute Error metric pinpoints the most significant deviation between measured or estimated values ($\hat{Y}$) and actual values (Y), providing insight into the worst-case performance scenario. The calculation is as follows:

$$\text{Max Error} = \max_{i=1}^{N} |\hat{Y}_i - Y_i| \tag{5.2}$$

Average Processing Time (APT)

The average processing time metric assesses the computational efficiency of each method by averaging the computational time used for each measurement step (T_s). The calculation is as follows:

$$\text{Average Processing Time} = \frac{1}{N}\sum_{i=1}^{N} T_{si} \tag{5.3}$$

5.1.3. Hardware Specifications

The simulations in this study were conducted on a personal computer to analyse various measuring solutions. The hardware configuration of the PC used for these simulations is as follows:

- Computer Model: Asus VivoBook N580GD
- Processor: Intel® Core™ i7-8750H Processor 2.2 GHz (9M Cache, up to 4.1 GHz)
- Number of CPU Cores: 6
- Memory (RAM): 16GB DDR4 SO-DIMM
- Graphics Processing Unit (GPU): NVIDIA® GeForce® GTX 1050 4GB GDDR5
- Storage: 1TB SATA 5400RPM 2.5” HDD, 256GB M.2 SATA SSD
- Operating System: Windows 11 Home (64-bit)

This hardware configuration proved to be sufficient for this study, providing the necessary computational resources for efficient analysis of the measuring solutions. This description can serve as a reference for implementing the algorithms in other systems with different specifications.

5.2. Measuring Solutions Verification

The measuring solutions underwent a verification process within the ideal simulation setup described in subsection 5.1.1. This verification process involved comparing the measurements obtained from different solutions to actual ground truth values, ensuring the correct implementation of the techniques. These findings are vital to identify inconsistencies or inaccuracies in the developed measuring methods. In an ideal scenario with perfect sensor parameters and an empty ship deck, errors are expected to be minimal. However, significant deviations from the actual values may signal potential issues with the developed techniques. The subsequent subsections will analyse the verification results for each method.

Deck Detection (LiDAR-MRU)

The *deck detection* method was applied using the LiDAR-MRU ideal sensor unit. The measurements were then compared to the true relative heave distance values(D).

In Figure 5.2, the top graph presents the acquired measurements and ground truth values. At the same time, the bottom plot provides the corresponding errors or deviations at each time step for further analysis. The MAE for the measurements was calculated at 0.005 meters, with a maximum error of 0.017 meters.

The verification test indicates that the deviations between the measured and actual values of D are minimal. While the use of ideal sensors would theoretically result in a MAE of zero, a perfect zero error was not achieved. Nevertheless, the random nature of the observed errors leads to the dismissal of an incorrect algorithm implementation. The source of these errors is attributed to either the iterative nature of the plane fit algorithm used for convergence to the best-fitting solution or inaccuracies in the pre-implemented sensor model. Nonetheless, the precision level for the *deck detection* (LiDAR-MRU) solution is estimated to be on the order of 10^{-2} meters.

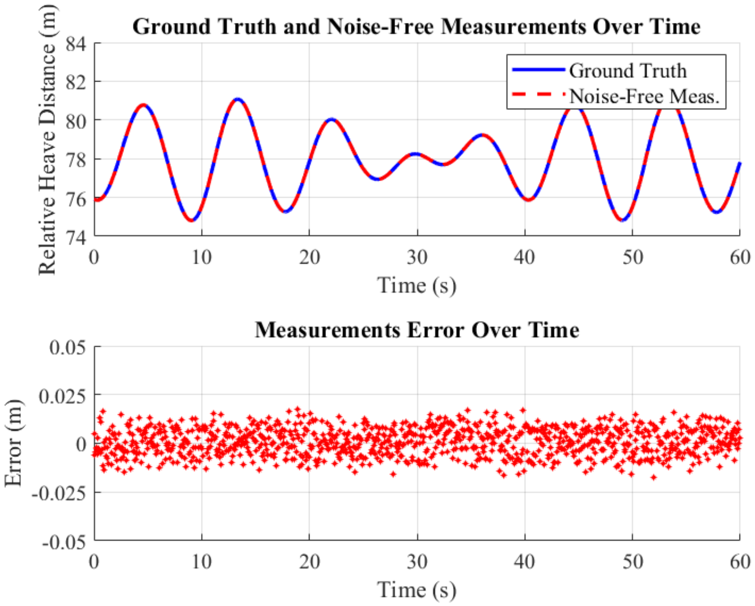


Figure 5.2: Verification of the *deck detection* method using the LiDAR-MRU sensor unit.

Deck Detection (RADAR-MRU)

The *deck detection* method was applied using the RADAR-MRU ideal sensor unit. The measurements were then compared to the true relative heave distance values(D).

In Figure 5.3, the top graph presents the acquired measurements and ground truth values. At the same time, the bottom plot provides the corresponding errors or deviations at each time step for further analysis. The MAE for the measurements was calculated at 0.00001 meters, with a maximum error of 0.0002 meters.

Despite employing the same measuring method as the previous section, the MAE aligns closely with the expected zero outcome when using the ideal RADAR-MRU sensor unit. The precision level for the *deck detection* (RADAR-MRU) solution is approximately on the order of 10^{-5} meters. These results imply that Simulink's LiDAR model may demonstrate a marginally lower degree of accuracy than its RADAR counterpart. It is important to stress that these sensors are used as black-box models, with the resulting errors being consistently minimal for the intended application, verifying the implemented *deck detection* method for measuring relative heave distance.

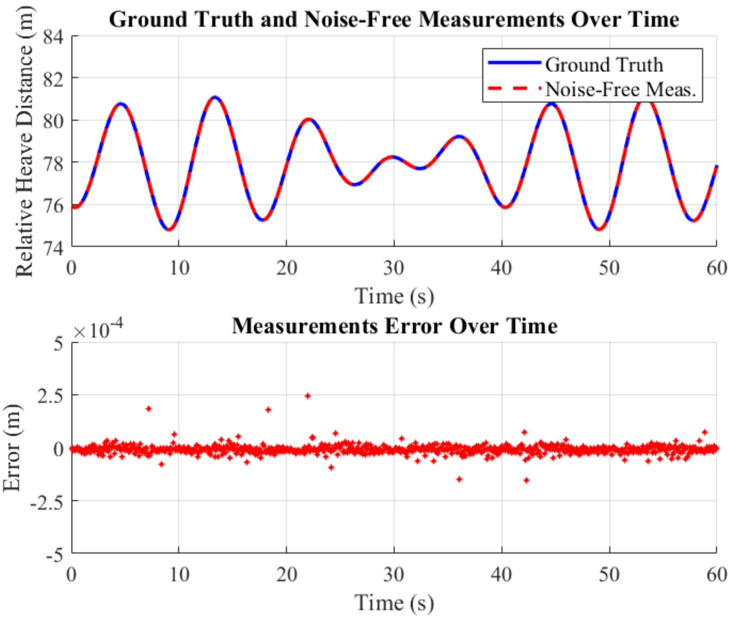


Figure 5.3: Verification of the *deck detection* method using the RADAR-MRU sensor unit.

Reflectors Detection (LiDAR-MRU)

The *reflectors detection* method was implemented solely with the LiDAR-MRU ideal sensor unit, and the obtained measurements were compared to the actual relative heave distance values (D).

In Figure 5.4, the upper graph displays the acquired measurements alongside the ground truth values. At the same time, the lower plot offers the associated errors or deviations at each time step for in-depth analysis. The MAE for the measurements was determined to be 0.010 meters, with a maximum error of 0.026 meters.

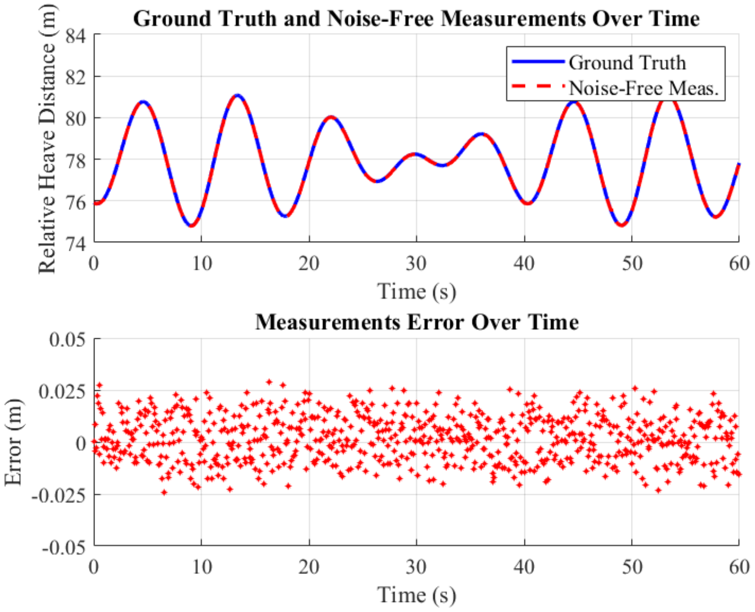


Figure 5.4: Verification of the *reflectors detection* method using the LiDAR-MRU sensor unit.

Once again, the observed errors in this solution appear to be random and of minor significance, verifying the intended functionality of the algorithm implementation. Although this solution shares the same

LiDAR model as the prior *deck detection* (LiDAR-MRU) approach, it exhibits variations in its results. The initial expectation was for the MAE and Max Error to align with the values obtained in the previous Figure 5.2. However, upon analysis, slightly higher error values were discerned. The main factor contributing to these discrepancies can be attributed to the processing stage, where a substantial number of data points are excluded due to their low reflectivity. This, in turn, impacts the accuracy of the plane fitting algorithm, as it relies on a smaller subset of points, leading to increased fitting errors.

Doppler-Based (RADAR-MRU)

The *Doppler-based* method was exclusively applied using the RADAR-MRU ideal sensor unit, and the collected measurements were compared with the actual relative heave speed values ($\dot{D}$).

The top graph on Figure 5.5 shows the acquired measurements and the ground truth values. On the other side, the lower plot provides the corresponding errors or deviations at each time step for comprehensive examination. The measurements' MAE was established at 0.018 meters per second, with a maximum error of 0.051 meters per second.

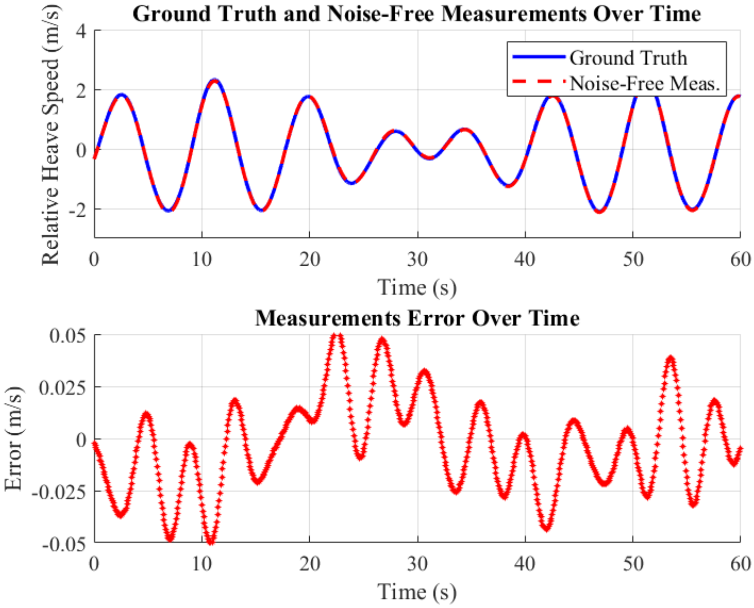


Figure 5.5: Verification of the *Doppler-based* method using the RADAR-MRU sensor unit.

In contrast to prior solutions, the current approach focuses on measuring the relative heave speed ($\dot{D}$). Notably, it deviates from the random error patterns observed in earlier verification tests. This deviation stems from a fundamental assumption made during the development of this measurement method, detailed in subsection 4.1.3, where it was assumed that rotational motions had negligible influence. As a result, the calculated error exhibits a periodic behaviour due to the effect of the existing rotational motions on relative heave. Despite a slightly higher error magnitude (on the order of 10^{-1}) compared to previous methods, it remains relatively low, supporting the validity of the assumption regarding the negligible impact of rotational motion in this specific context.

Despite the previous test, no conclusive evidence regarding the absence of issues within the implemented measuring solution can be established. A distinct verification case was conducted to address this, preserving the ship's translational movements while excluding roll, pitch, and yaw. In this scenario, an error close to zero is anticipated, as the assumption of no rotational motion should not affect the calculated error. The outcomes of this verification case are detailed in Figure 5.6.

In this verification case, the MAE for the measurements stands at 0.005 meters per second, with a maximum error of 0.008 meters per second. The expectation was for the error to diminish compared to the previous test, approaching zero and becoming non-periodic. While there is a noticeable reduction in the error, now on the order of 10^{-2}, periodicity persists, initially attributed to potential modelling inaccuracies in the solution. A comprehensive algorithm examination revealed no issues. The author concluded that the error stems from a slight delay in the obtained speed measurements from Simulink's RADAR Model. Nevertheless, these errors are of minimal significance, and the verification affirms the correct implementation of the method.

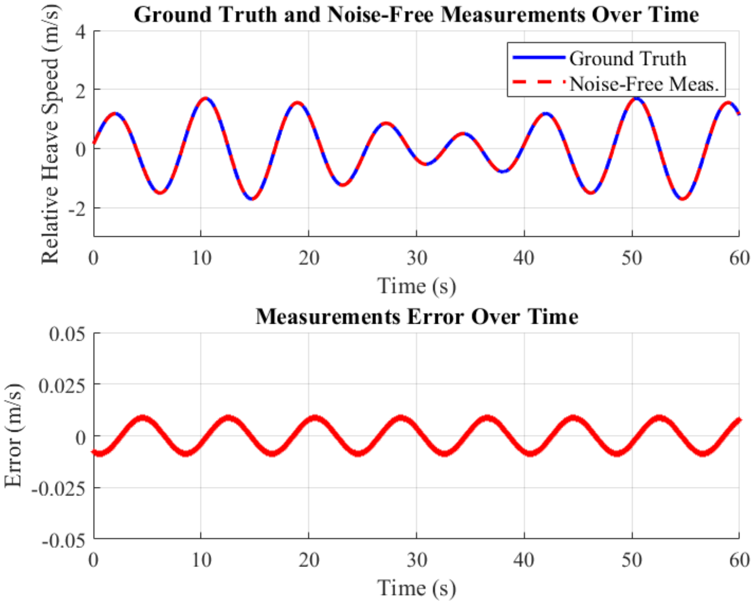


Figure 5.6: Verification of the *Doppler-based* method using the RADAR-MRU sensor unit, with the exclusion of rotational ship motions.

5.3. Measuring Solutions Evaluation

The preceding section effectively verified the implemented solutions, confirming their alignment with the expected results. However, thoroughly evaluating and comparing these measuring solutions under realistic conditions and with less-than-ideal sensors remains essential to draw conclusive results about the methods' applicability. This section is dedicated to assessing and comparing the solutions, with a specific focus on accuracy and processing speed. Moreover, a sensitivity study is conducted to analyse how various parameters impact the performance of the implemented solutions.

5.3.1. Performance Analysis

The measuring solutions evaluation begins with analysing the accuracy of relative heave distance estimates, portrayed in Figure 5.7, with the evaluation setup detailed in subsection 5.1.1. The figure's upper panel displays the $\hat{D}$ error distributions, offering insights into the errors' magnitude and standard deviation. At the same time, the lower panel presents the Mean Absolute Error and Maximum Absolute Error for each implemented solution. These values are normalised to the maximum amplitude of the ground truth reference (depicted in Figure 5.1), facilitating a more explicit interpretation of the results. Notably, for the *Doppler-based* method, the initial D state is assumed to be known, allowing for a more straightforward comparison with the results obtained by other methods.

Based on the findings presented in Figure 5.7, it is evident that the *deck detection* method employing the LiDAR-MRU sensor unit stands out as the most accurate (close to the true values) and precise solution, having the lowest normalised MAE of 0.3% (or $0.010[m]$ in absolute value) and a normalised Max Error of 1.4% (or $0.045[m]$).

While the *reflectors detection* method with the LiDAR-MRU unit achieved the second lowest MAE of 0.4% (or $0.013[m]$), it is important to note that the Max Error increased to 2.9% (or $0.088[m]$) compared to the previous method. This increase suggests that, despite its similar accuracy, relying solely on reflectors may compromise robustness due to the reduced number of points used and could lead to occasional outlier measurements. This phenomenon can be attributed to potential obstructions of the reflectors caused by objects placed on the deck.

The *deck detection* method using the RADAR-MRU unit achieved the third-best MAE of 0.6% (or

$0.018[m]$). The disparity in performance compared to its LiDAR-MRU counterpart is primarily attributed to the inherent limitations of RADAR range measurements, which are less accurate. Notably, the higher Max Error of 4.2% (or $0.128[m]$) cannot be solely explained by RADAR technology's inferior range and angular resolution. The reduced FOV and sparse point cloud resulting from RADAR measurements also contribute to its decreased robustness.

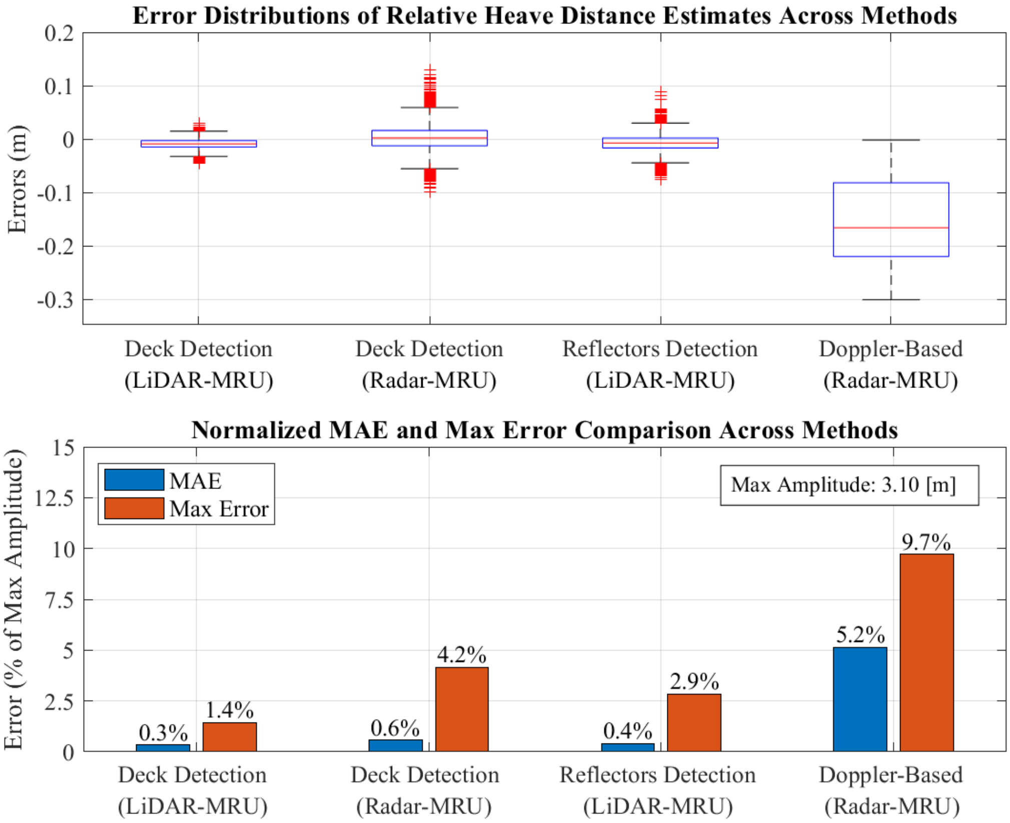


Figure 5.7: Performance analysis of relative heave distance estimation across the implemented solutions. The upper panel displays error distributions, while the lower panel compares the computed MAE and Max Error.

The presence of a substantial number of outliers (depicted as red stripes) in both the *deck detection* (RADAR-MRU) and *reflectors detection* (LiDAR-MRU) methods is indicative of their higher Max Errors when compared to the *deck detection* approach employing the LiDAR-MRU unit. However, it is worth noting that despite the higher Max Errors, the Mean Absolute Errors among these three solutions differ by only a few millimetres.

The *Doppler-based* method, employing the RADAR-MRU sensor unit, yields the highest MAE (5.2%, or $0.159[m]$) and Max Error (9.7%, or $0.301[m]$). These increased errors are mainly attributed to the method's reliance on relative heave speed measurements to derive the relative heave distance estimates. Neglecting rotational motion influence and inherent speed measurement errors results in error accumulation over time. Even within the 60-second simulation period, the drift effect becomes evident, with the median of the error distribution ($0.166[m]$) deviating from zero, in contrast to the error distributions of other methods. Consequently, this method's relative heave distance estimates are most reliable within shorter time intervals, where the impact of drift is negligible.

In the subsequent phase of evaluating the measuring solutions, the accuracy of relative heave speed estimates is depicted in Figure 5.8. The upper panel of the figure illustrates the error distributions for $\hat{\dot{D}}$, while the lower panel presents the MAE and Max Error values for each of the implemented solutions. Upon close examination of Figure 5.8, a notable trend emerges: the speed estimation errors have increased across all solutions, except for the *Doppler-based*. This observation aligns with the fact that the *deck detection* and *reflectors detection* methods no longer directly measure speed but instead rely solely on distance measurements to estimate it.

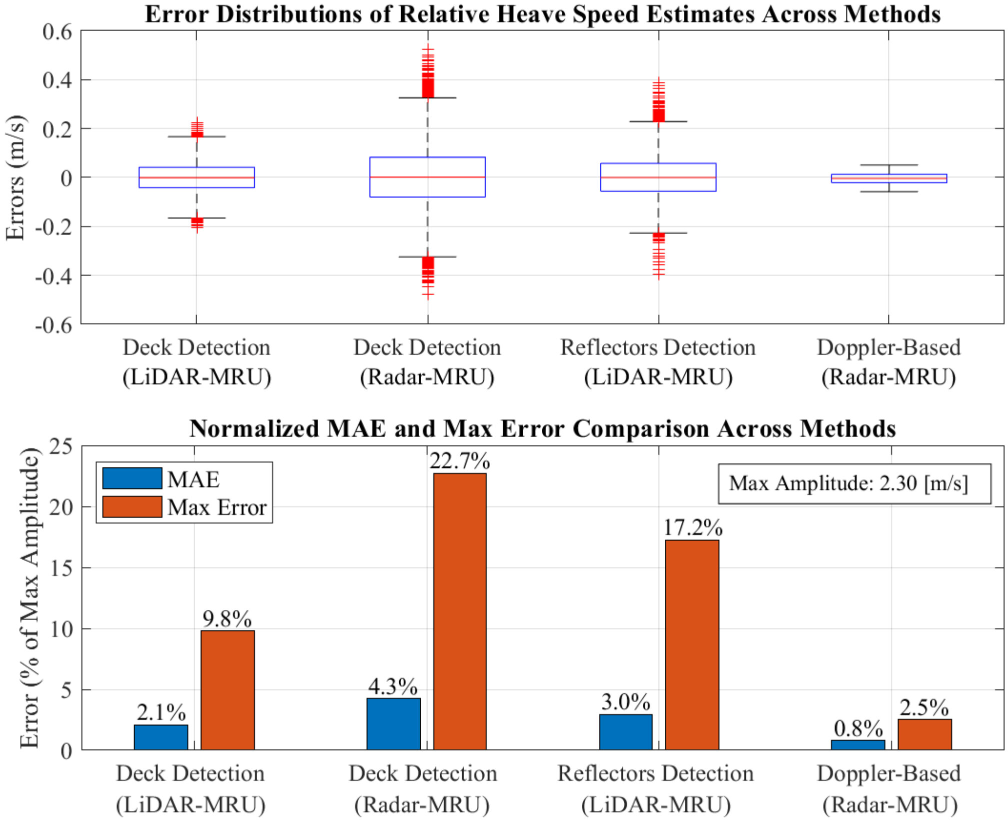


Figure 5.8: Performance analysis of relative heave speed estimation across the implemented solutions. The upper panel displays error distributions, while the lower panel compares the computed MAE and Max Error.

The most promising results were achieved with the *Doppler-based* method, having a normalised MAE of 0.8% (or $0.018[\frac{m}{s}]$ in absolute value) and a normalised Max Error of only 2.5% (or $0.057[\frac{m}{s}]$). This outcome underscores the advantages of directly measuring speed, a capability not shared by the other methodologies.

As speed estimates for the *deck detection* and *reflectors detection* methods rely on distance measurements, the ranking of these methods closely resembles that observed in the previous Figure 5.7, albeit with higher errors due to the amplification of the distance estimate errors. Specifically, the *deck detection* MRU-LiDAR method yields the second-best overall MAE of 2.1% (or $0.048[\frac{m}{s}]$) and a Max Error of 9.8% (or $0.225[\frac{m}{s}]$). In comparison, the *deck detection* RADAR-MRU method achieves an MAE of 4.3% (or $0.098[\frac{m}{s}]$) and a Max Error of 22.7% (or $0.522[\frac{m}{s}]$). The *reflectors detection* method performs in a similar range, obtaining an MAE of 3.0% (or $0.068[\frac{m}{s}]$) and a Max Error of 17.2% (or $0.396[\frac{m}{s}]$).

To conclude the performance analysis, a comparison of processing speed is provided for all implemented solutions, presenting the Average Processing Time for each measuring cycle in Figure 5.9. It is crucial to emphasise that, depending on the specific application, the algorithm's processing time can significantly impact the overall accuracy of the implemented solution. A sensitivity study is presented in subsection 5.3.2, examining various factors influencing the methodologies' performance, including time delay. Ideally, one would seek the fastest and most accurate solution. However, it is often necessary to strike a balance between processing speed and accuracy, as improvements in one aspect may come at the expense of the other.

As evident in Figure 5.9, a noticeable distinction emerges between the *deck detection* method and the others. The *reflectors detection* and *Doppler-based* methods stand out with significantly faster processing times. Conversely, the *deck detection* method, which entails an iterative process of fitting a plane using a large set of data points, exhibits lower efficiency due to its iterative nature.

The difference in processing speed between the LiDAR-MRU and RADAR-MRU *deck detection* methods can be attributed to the sparsity of the RADAR point cloud, resulting in fewer data points and faster execution of the plane fitting algorithm. The LiDAR-MRU *deck detection* method could be expedited by further reducing the number of data points in the downsampling stage. However, reducing the processing time can come at the cost of compromising its superior accuracy. The sensitivity study in subsection 5.3.2 shows the impact of the downsampling factor and how it can be optimised for the specific application requirements.

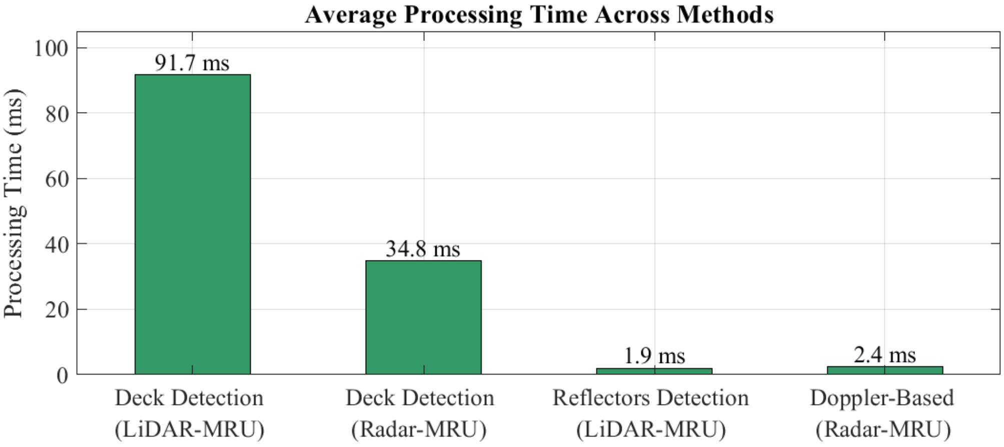


Figure 5.9: Performance analysis of processing speed across the implemented solutions.

5.3.2. Sensitivity Study

The preceding evaluation section presents a comprehensive performance comparison of the solutions. However, it is crucial to consider other factors relevant to practical implementation. This section conducts a sensitivity study, examining the influence of various factors on the solutions' performance to provide a more thorough understanding of their overall effectiveness.

This sensitivity study delves into several critical aspects. **Time Delay** analyses the influence of time delay in measurements on the estimation error. **Motion Reference Unit (MRU)** investigates the impact of MRU measurement errors in the implemented solutions. **Measurement Frequency** discusses the optimal measurement frequency. **Point Cloud Density** analyses the effects of the exteroceptive sensor's point cloud density. Lastly, **Reflective Noise** briefly discusses how incorrect reflective points can affect the performance of the *reflectors detection* method.

Time Delay

Time delays within the measuring system can originate from multiple sources, such as communication delays, processing time, and sensor dynamics. Consequently, comprehending how various design choices within the system can affect time delays and influence overall measurement errors is essential for effectively implementing these solutions. The product between relative heave speed and the time delay was used to quantify the expected distance error for each measurement resulting from time delays, as expressed in Equation 5.4.

$$\text{Distance Measurement Error} = \text{Actual Speed} \times \text{Time Delay} \tag{5.4}$$

Figure 5.10 provides a qualitative and quantitative representation of this estimation error. For instance, in sea conditions leading to an average relative heave speed of $1[\frac{m}{s}]$ and a measurement delay of $100[ms]$, the relative heave distance measurement error is as significant as $0.100[m]$. The magnitude of this error is considerable and should be a central consideration when implementing the relative heave measurement system. It stresses the importance of addressing time delay as one of the primary sources of error within the measuring system.

While this analysis specifically pertains to the methods directly measuring D, it is crucial to note that in the case of the *Doppler-based* method, similar speed measurement errors can be expected by considering acceleration rather than speed.

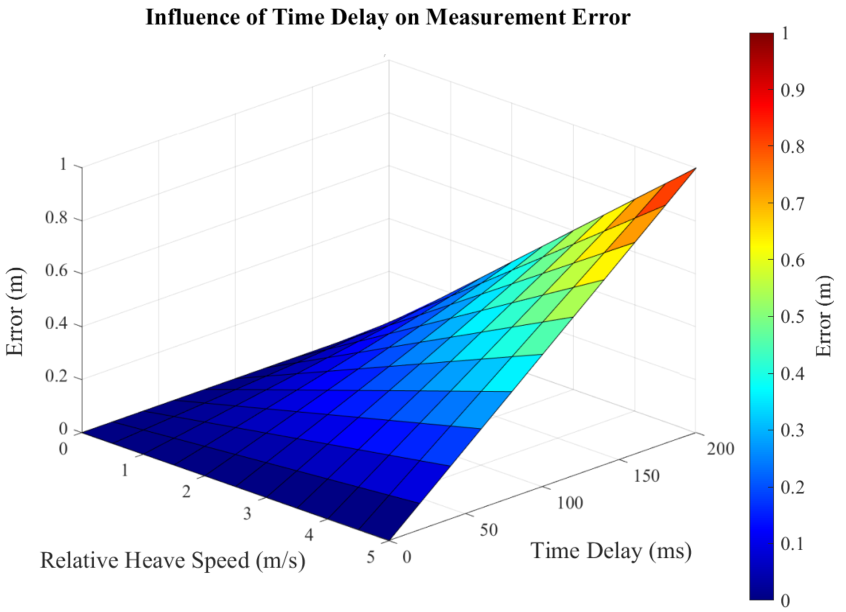


Figure 5.10: Influence measurement delays on relative heave distance errors, contingent on the relative heave speed.

Motion Reference Unit (MRU)

The Motion Reference Unit plays a crucial role in the measuring system. Its measurements enable the conversion of point clouds captured in the exteroceptive sensor frame into a global coordinate system with a fixed orientation. Thus far, MRU measurements have been assumed to be ideal, meaning that the previously mentioned errors are independent of any MRU measurement inaccuracies. Although detailed modelling and analysis of MRU sensors is beyond the scope of this thesis, a brief analysis of their potential influence on measurement errors has been conducted. This sensitivity study is essential to assess MRU requirements and expected measurement errors when implementing this solution under actual conditions.

In the subsequent Figure 5.11, the left plot illustrates the expected distance measurement errors for the *deck detection* and *reflectors detection* methods relative to the standard deviation of MRU measurement errors. On the right side, the plot displays the speed measurement errors for the *Doppler-based* method.

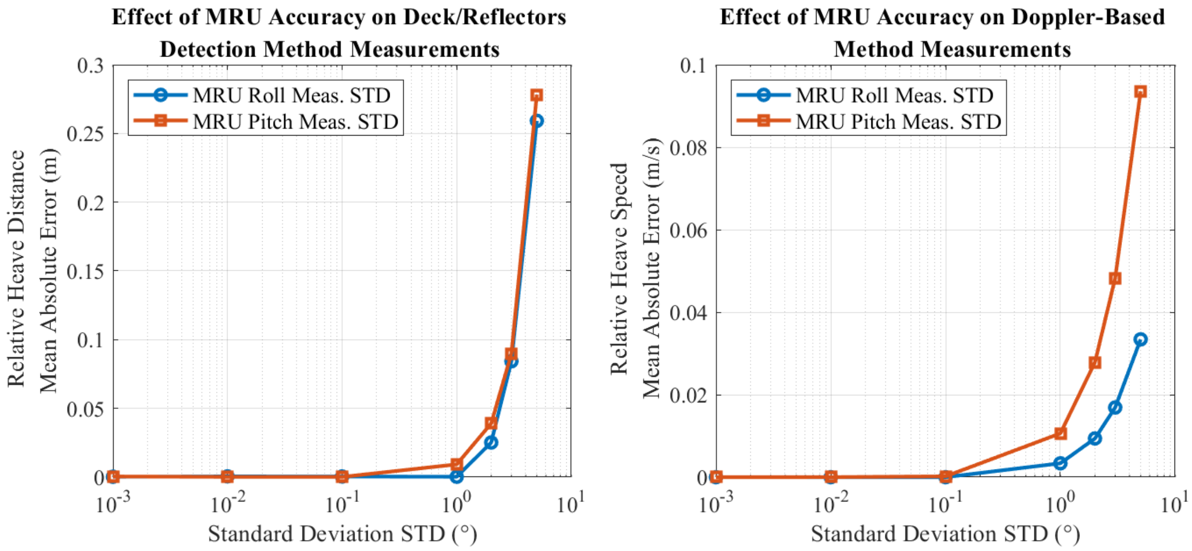


Figure 5.11: Influence of MRU measurement errors, particularly roll (in blue) and pitch (in red), on the measurement methods. On the left, it illustrates the D errors for the *deck detection* and *reflectors detection* methods (only *reflector detection* errors were analysed since MRU use is identical). On the right, the plot displays the $\dot{D}$ estimate errors for the *Doppler-based* method.

MRUs offer precision levels that can extend down to 0.001 degrees [28]. Examining the previous figure, it becomes evident that the high accuracy of MRUs results in a negligible influence on the overall measurement system. However, it is necessary to acknowledge that MRU measurements rely not only on sensor accuracy but are also influenced by the physical distance from the exteroceptive sensor. In real-world scenarios, the flexibility of the crane's structure can introduce some degree of error. Nevertheless, if the overall MRU measurement accuracy remains within the 1-2 degrees range, it should still lead to relatively low measurement errors. If the boom's flexibility is excessively high, an alternative option is to position the MRU adjacent to the exteroceptive sensor.

Measurement Frequency

In signal processing, the Nyquist-Shannon Sampling Theorem is a fundamental principle. It states that the sampling frequency should be at least twice the highest component frequency within the signal to avoid signal distortion or aliasing [104].

In the simulation, the highest frequency component of the reference signal operates at approximately $0.15[Hz]$. Thus, a minimum sampling of $0.3[Hz]$ is necessary to adhere to the Nyquist-Shannon Sampling criterion. Nonetheless, using a sampling frequency ten times higher than the minimum frequency component is standard practice to ensure robust signal representation. Hence, an appropriate sampling frequency for this scenario would be around $1.5[Hz]$. However, this principle does not guarantee that errors induced by the measurement delay will remain within the desired standards.

The Kalman Filter estimation algorithm can handle high sampling frequencies for a low number of system states without imposing significant computational costs. Therefore, opting for small sampling frequencies does not offer any advantages. However, the errors in the estimation process highly depend on the sensor's measurement frequency. These two frequencies do not necessarily have to match in the estimation algorithm.

A sensitivity study was conducted to assess the influence of the sensor's measurement frequency on relative heave estimate errors and determine the optimal sensor measurement frequency for each developed method. The findings from this study are presented in Figure 5.12.

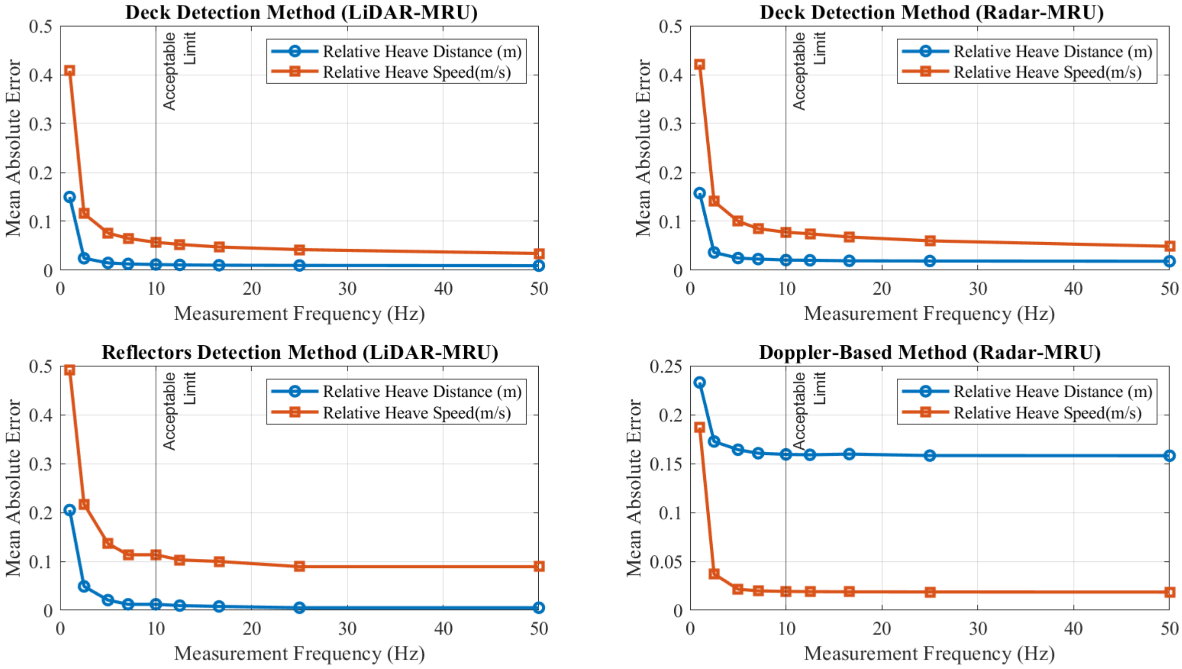


Figure 5.12: Influence of measurement frequency on relative heave distance (blue) and speed (red) estimation errors across the implemented solutions.

Upon examining Figure 5.12, it is apparent that once the measurement frequency exceeds $10[Hz]$, the error shows minimal variation, suggesting the preference for a sensor with a $10[Hz]$ or higher measurement capacity. Although, lower measurement speeds would be acceptable for slower ship motion scenarios and vice versa.

Point Cloud Density

The point cloud density depends on the angular resolution of the chosen sensor. Hence, it is a crucial parameter to consider when selecting a sensor, as it directly influences the measuring solutions' accuracy. In the case of the *Doppler-based* method, point cloud density is less critical. This algorithm can function with a relatively low-density point cloud, as demonstrated by its functionality with just three detections. However, a higher point cloud density can enhance its robustness, particularly in mitigating inaccurate Doppler measurements.

On the other hand, the *reflectors detection* method is significantly influenced by point cloud density. The ideal point cloud density will depend on the size and placement of the reflectors. An optimal point cloud density should ensure multiple detections for all reflectors within a direct line of sight.

Lastly, for the *deck detection* method, obstacles on the deck can impact its performance. Thus, it is essential to determine the point cloud density at which results begin to deteriorate. The following Figure 5.13 provides insight into how the performance of *deck detection* with the LiDAR-MRU sensor unit varies with changes in point cloud density.

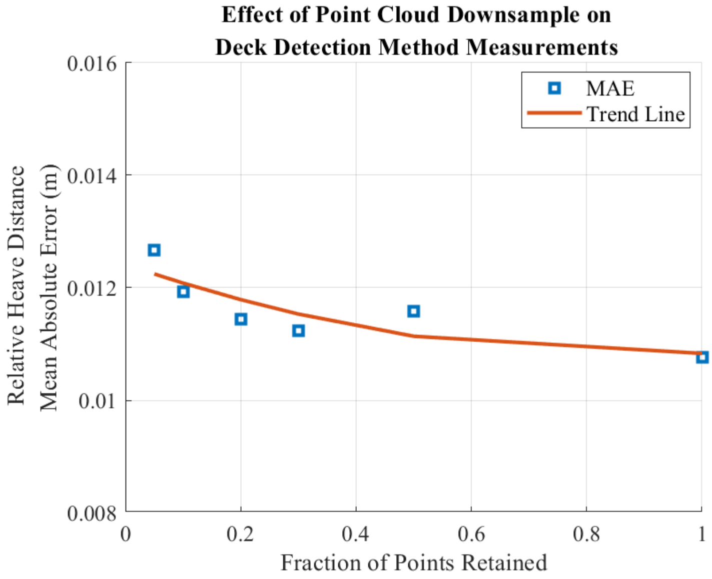


Figure 5.13: Influence of point cloud density on the relative heave distance measurement of the *deck detection* method.

It is worth noting that the previous Figure 5.13 indicates only a slight change in the measurements MAE with a reduction in the total number of points. This observation suggests that high resolution is not a critical factor to consider when selecting a sensor for the *deck detection* method. This outcome aligns with the objective of detecting the largest geometry in view: the ship deck. In this context, sensor resolution does not play a significant role. However, resolution would become more critical if the goal were to distinguish between objects placed on the deck.

Considering that point cloud density and sensor resolution are not the primary concerns, one might wonder why the RADAR-MRU sensor performs less effectively than LiDAR. The answer lies in the Field Of View. Even when downsampling the LiDAR point cloud to a similar density, its superior coverage makes it well-suited for *deck detection*. Therefore, when selecting the FOV of a 3D sensor, it is essential to ensure that the coverage encompasses most of the deck area, striking the right balance and not exceeding or falling short.

Reflectivity Noise

For the *reflectors detection* method, an intensity threshold is established, considering the anticipated high reflectivity of the markers located on the supplier's deck. However, a question arises regarding the possibility of non-marker reflections surpassing this threshold. To address this concern, an analysis was conducted to assess the impact on the method's performance.

In Figure 5.14, the graph illustrates the variation of relative heave distance MAE in response to the presence of random false reflective points, which are not associated with the markers.

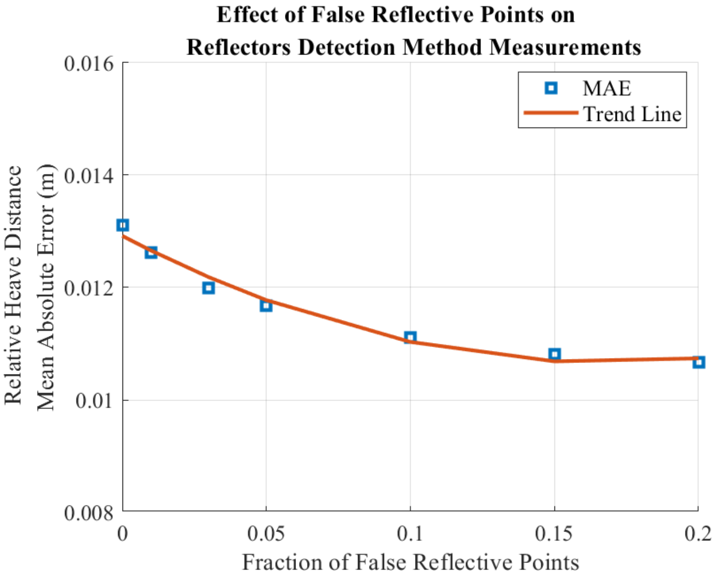


Figure 5.14: Influence of false reflective points on relative heave distance (D) measurements in the *reflectors detection* measuring solution.

As illustrated in the preceding Figure 5.14, an intriguing phenomenon emerges: as the proportion of false reflective points increases, the Mean Absolute Error decreases, approaching the MAE achieved by the *deck detection* method. This result is logical, as it is akin to reducing the threshold filter to the point where all data points are accepted. It is worth noting that this transition not only causes the MAE of the measurements to approach that of the *deck detection* method but also has a similar effect on processing speed, as a larger number of points are considered for the plane fit algorithm.

5.4. Discussion

The primary aim of this chapter is to conduct a comparative analysis to determine the optimal approach for measuring relative heave. The chapter starts by providing an overview of the simulation setup. Subsequently, each solution undergoes a verification process to assess its correctness and capability to measure relative heave accurately. Following this verification, the measuring solutions are thoroughly evaluated, including a performance and sensitivity analysis addressing **Research Questions 6 and 7**.

An overview of the results achieved with each implemented measurement solution is presented in Table 5.4. A thorough analysis of these results reveals the absence of a singular optimal solution, but rather, the existence of multiple viable alternatives for implementation. However, certain well-founded conclusions can be drawn, indicating which alternatives may be more suitable in specific contexts.

The *reflectors detection* method offers an optimal trade-off between accuracy and processing speed. Its efficiency in expediting data processing through reflectivity-based point filtering minimises time delay errors, and it ranks as the second most accurate method in terms of Mean Absolute Error. It is worth noting that the effectiveness of this method relies on the installation or temporary placement of reflectors on the feeder vessel, which can sometimes be a limitation in practical situations. Nevertheless, it has proven to be robust in dealing with false reflective points that may occur in real-world scenarios.

Conversely, the *deck detection* method proves to be the ideal choice for a reflector-independent measurement solution. It offers both robust and accurate relative heave measurements across all types of feeder vessels with a discernible ship deck. However, it does require a careful trade-off between accuracy and processing speed. The optimal selection of the plane fit confidence is key, as a significant increase in accuracy may come at the cost of processing speed. While the *deck detection* method performed adequately with the RADAR-MRU sensor unit, choosing the LiDAR-MRU sensor unit is advisable. The latter provides enhanced coverage of the supplier vessel, a crucial factor for achieving higher accuracy. While it does come with a longer processing time, this drawback can be mitigated through hardware improvements or optimising the accuracy-speed trade-off.

Lastly, the standalone use of the *Doppler-based* method is discouraged, as it cannot measure the absolute relative heave distance, resulting in potential drift issues, even over brief periods. Nonetheless, it offers a significant advantage by delivering the most accurate relative heave speed measurements.

Within the scope of this thesis, each solution has been assessed individually. Nevertheless, it is worth considering the potential benefits of their integration. Notably, while the standalone use of the *Doppler-based* method is not recommended, its integration with other methods is promising to enable concurrent distance and speed measurements, improving the performance of real-time filtering algorithms. For instance, using the RADAR-MRU sensor unit and integrating *deck detection* with *Doppler-based* measurements could mitigate the accumulation of drift errors while allowing for highly accurate distance and speed estimates. This concept can be extended to a LiDAR-RADAR-MRU sensor unit, using the LiDAR for improved *deck detection* accuracy.

Future advancements in the industry may lead to the emergence of Doppler LiDAR [60, 61], which holds promise as the optimal sensor solution. Nevertheless, it is essential to acknowledge that this technology is still under development. Additionally, 3D imaging RADARs [70, 71], although more advanced, only start production in 2024 by some brands, so there is also some risk involving tender technology.

Table 5.4: Implemented measuring solutions performance overview

KPI	MAE_D		$MAE_{\dot{D}}$		APT
	$[m]$	[%]	$[\frac{m}{s}]$	[%]	$[ms]$
1. Deck Detection (LiDAR-MRU)	0.010	0.3	0.048	2.1	91.7
2. Deck Detection (RADAR-MRU)	0.018	0.6	0.098	4.3	34.8
3. Reflectors Detection (LiDAR-MRU)	0.013	0.4	0.068	3.0	1.9
4. Doppler-Based (RADAR-MRU)	0.159	5.2	0.018	0.8	2.4

Note: The percentages, [%], indicate the proportion relative to the maximum amplitude of the relative heave ground truth.

6 Concluding Remarks

The concluding chapter of this study begins by addressing the initially proposed sub-questions in section 6.1. These responses ultimately converge to address the main research question:

How can novel feeder-independent measuring methods be developed and assessed for relative heave estimation during offshore Ship-to-Ship cargo transfer operations?

Following this, section 6.2 offers recommendations for future research, outlining the necessary steps for improving and implementing the developed relative heave measurement solutions in practice.

6.1. Addressing Research Questions

1. What are the current limitations of relative heave measurement systems?

Some pioneering maritime crane companies are now offering Relative Heave Compensation systems. These systems are designed to mitigate the heave motion between two ships, also known as relative heave, caused by the sea conditions. A prevalent industry practice for measuring relative heave involves using a Motion Reference Unit on both the crane and the feeder vessel. Each MRU sensor records its respective body motions, and a radio communication link transmits this data from the feeder vessel to the crane vessel, where the relative heave is computed. However, the implementation of this measuring system has faced resistance from crane vessel owner companies due to two primary challenges. Firstly, MRU sensors are often lacking on feeder vessels, and secondly, complications arise from the bureaucratic hurdles associated with sharing information among vessels owned by different companies. Additionally, when the MRU sensor is unavailable, the need to place temporary sensors on the supplier vessels further complicates Ship-to-Ship (STS) transfer operations. Hence, there is a growing interest in developing a feeder-independent relative heave measurement system.

2. What sensor units provide a robust and supplier-independent alternative for relative heave measurement?

Existing research mainly explored the use of two MRUs, one on each ship, and a third exteroceptive sensor to measure the absolute distance between the ships. This third sensor also provided position and attitude measurements of the second vessel. Noteworthy research conducted by Tørdal [11] concluded that the MRU on the feeder vessel could be entirely replaced by a 3D sensor or a time-of-flight sensor capable of detecting and tracking the feeder vessel's movements. Remarkably, only one study [80] proposed a feeder-independent solution involving an MRU-Camera-LiDAR sensor unit. However, using a camera sensor as the primary sensor for ship detection is not ideal, particularly for long-distance and offshore applications. Moreover, it requires calibration between the camera and LiDAR sensors.

Drawing upon existing literature, this study proposes a sensor unit comprising a Motion Reference Unit (MRU) and a 3D exteroceptive sensor for relative heave measurement. Based on analogous applications within the automotive industry, two potential 3D exteroceptive sensors emerged as viable choices, namely the imaging RADAR and LiDAR sensors. Instead of making an exclusive choice, this research delves into both the LiDAR-MRU and RADAR-MRU sensor units due to their shared characteristics.

3. How can a kinematic model be formulated to represent relative heave during Ship-to-Ship operations, accounting for all six degrees of freedom of both vessels?

The study has determined that modelling relative heave in Ship-to-Ship cargo transfer operations can be achieved using a vector diagram. This approach accounts for the six degrees of freedom inherent to each vessel, encompassing heave, surge, sway, roll, pitch, yaw, and their respective derivatives. The investigation resulted in the formulation of two key expressions. The first equation, consisting of 16 terms, computes the relative heave distance, considering both vessels' positions and attitudes and the crane's geometry. The second equation, composed of 19 terms, computes the relative heave speed, factoring in the linear and angular velocities of the vessels and their respective attitudes. While some expressions could have been simplified by omitting negligible terms, they were retained in their complete form to ensure a precise assessment of the measurement solutions.

4. How can simulated sensor data be generated to support the development of a relative heave measuring system in the context of Ship-to-Ship operations?

Due to the unavailability of actual sensor data, this study established a simulation framework for generating synthetic point cloud data emulating Ship-to-Ship cargo transfer operations. A co-simulation was implemented using Simulink and Unreal Engine software. Unreal Engine enabled the design of a realistic 3D scene featuring two ships and a crane to mimic the operation. The Simulink model used the built-in LiDAR and RADAR sensor models to extract real-time data while setting the ships' motions within the 3D environment. Concurrently, the positions and orientations of the crane vessel were recorded, considering the MRU sensor. Furthermore, ground truth values were computed for each time step using established mathematical expressions and recorded for assessment purposes.

5. How can measuring methods be developed to process and analyse the simulated data for real-time relative heave estimation?

This study introduces a range of solutions for measuring relative heave, encompassing three essential phases: data processing, data analysis, and estimation. Three distinct measurement methods have been developed: *deck detection*, *reflectors detection*, and *Doppler-based*, each with a unique data processing and analysis methodology. As a result, four distinct measurement solutions were implemented: *deck detection* (LiDAR-MRU), *deck detection* (RADAR-MRU), *reflectors detection* (LiDAR-MRU), and *Doppler-based* (RADAR-MRU).

The *deck detection* approach utilises position data from all reflected points, ensuring a comprehensive coverage of the feeder vessel deck. A model-based approach is then employed to fit an appropriate plane model to the feeder vessel deck, facilitating the computation of relative heave distance.

The *reflectors detection* method, exclusively used with the LiDAR-MRU sensor unit, aims to derive a suitable plane equation to track the feeder vessel deck. However, it exclusively utilises points associated with retro-reflective markers placed on the feeder vessel, detected based on a reflectivity threshold.

The *Doppler-based* method, implemented with the RADAR-MRU unit, leverages Doppler velocities measured for each point within the point cloud. It uses these velocity measurements and a correlation function to determine the optimal fitting relative heave speed.

Following the data processing and analysis phase, the measurements obtained from each implemented solution undergo an estimation process. Within this stage, a Kalman Filter is used to enhance these measurements and derive estimates for both relative heave distance and speed.

6. How does the performance of the implemented measuring solutions compare against each other, considering accuracy and processing time?

The *reflectors detection* method, utilising the LiDAR-MRU sensor unit, achieves an optimal balance between accuracy and speed, yielding a Mean Absolute Error normalised to the ground truth maximum amplitude of 0.4% for relative heave distance and 3.0% for speed estimates. It accomplishes this while maintaining an average measurement cycle time of $1.9[ms]$.

In cases where the placement of reflectors on the feeder vessel is not feasible, the *deck detection* method employing the LiDAR-MRU presents a viable alternative. It demonstrates impressive results with a normalised MAE of 0.3% for relative heave distance and 2.1% for speed estimates. However, it requires a longer processing time of $91.7[ms]$ per measurement cycle, necessitating careful consideration of the trade-off between speed and accuracy within the context of implementation requirements. While the *deck detection* method utilising a RADAR-MRU unit achieves good results, the LiDAR-MRU sensor unit emerges as a more suitable option due to its superior coverage and range accuracy.

On the other hand, the *Doppler-based* system using a RADAR-MRU sensor unit is not recommended as a standalone solution due to drift issues in the relative heave distance estimates. Nevertheless, it exhibits remarkable relative heave speed estimates with a normalised MAE of 0.8% and a processing time of $2.4[ms]$ per measurement cycle, making it a promising candidate for integration with any of the previously mentioned methods to achieve optimal performance.

7. What factors can influence the performance of the developed measurement methods?

Sensor dynamics, communication delays, and processing time result in time delays within the measuring system. The compound effect of these time delays has been identified as a significant parameter affecting the performance of the measuring solutions. Notably, with a measurement delay of $100[ms]$ and relative heave speed of $1[\frac{m}{s}]$, the relative heave distance measurement error is as substantial as $0.100[m]$.

Regarding sensor characteristics, it was determined that MRU measurement errors do not significantly impact the measuring solution's performance because industrial-grade MRU angular errors are not significant enough. Furthermore, for the exteroceptive sensors (RADAR or LiDAR), the density of the point cloud was found to have minimal relevance for both the *Doppler-based* and *deck detection* methods. However, the Field Of View was identified as a crucial factor in all methods. Therefore, selecting a sensor with a complete coverage of the ship deck to be detected was deemed essential. Another consideration in sensor selection is measuring frequency. It was observed that a measuring frequency exceeding $10[Hz]$ does not yield a substantial improvement, a capability readily provided by current LiDAR and RADAR sensors.

Lastly, concerning the detection of reflectors, false reflective points were considered a potential issue. Nonetheless, it was determined that these false points do not significantly affect the solution's performance.

After addressing all relevant research questions, a comprehensive answer can be drawn for the main research question:

The study concluded that integrating 3D sensors, such as RADAR and LiDAR, with the MRU measurements of the crane vessel is a highly suitable approach for independently measuring relative heave distance and speed during offshore Ship-to-Ship cargo transfer operations. A specialised simulation environment was established for prototyping innovative measurement solutions, providing the foundation for developing data processing and analysis methods capable of handling the acquired data to measure pertinent system states. An estimation filter emerged as the optimal method for obtaining robust real-time estimates of all necessary system states for Relative Heave Compensation.

The implemented solutions have demonstrated their potential for industry practice, combining robust sensors and achieving accurate results. The achieved Mean Absolute Errors for relative heave distance estimates ranged from 0.3% to 5.2% of the reference's maximum amplitude, and speed estimates demonstrated accuracy within the range of 0.8% to 4.3%. Notably, these solutions exhibit relatively low computational times, spanning from 1.9 to 91.7 milliseconds, making them promising candidates for real-time application. Among these methods, the *reflectors detection* approach using LiDAR-MRU strikes an optimal balance between accuracy and processing speed, though it relies on reflective markers. The *deck detection* method, also utilising LiDAR-MRU, offers the highest accuracy in distance measurements but operates at a slower processing speed. In contrast, the *Doppler-based* method with Radar-MRU excels in speed measurements but does show vulnerability to distance estimation drift.

6.2. Recommendations for Further Research

This research used in-built sensor models from Simulink and extensively verified the results obtained for the different implemented measuring solutions. However, actual sensor data obtained on an up-to-scale test setup is essential to validate the acquired simulation results. Therefore, the author suggests testing the developed measuring solutions and sensor units in a setup similar to the one used by Tørdal [11] as the next research step to reach the technical readiness level for industry practice. Moreover, conducting experimental tests to assess the robustness of LiDAR and RADAR sensors to varying weather conditions is also important. Even though the feeder vessel's surrounding sea is expected to not significantly impact the point cloud data, a deeper investigation is still needed.

An additional research avenue involves optimising the chosen parameters in the data processing, analysis, and estimation algorithms specific to each measurement solution to improve overall performance. For instance, when implementing the *deck detection* method, a comprehensive evaluation of the trade-off between accuracy and processing speed is recommended. Furthermore, potential research directions include exploring completely alternative processing algorithms and advanced estimation techniques to reduce the negative impact of outlier measurements. It is worth noting that the developed software is intentionally structured in a modular manner, allowing for the prompt implementation of alternative processing and estimation algorithms.

This study intentionally refrained from fusing multiple measuring methods. Nevertheless, concurrently implementing these methods holds great promise for generating hybrid solutions that leverage each individual strengths. Therefore, it is highly advisable to investigate this avenue, for instance, fusing the *Doppler-based* solution with the others or combining the *deck detection* and *reflectors detection* methods to improve robustness.

The author considers the RADAR-MRU and LiDAR-MRU sensor units highly suitable for the problem at hand. However, future research should explore optimal configurations for these sensors on the crane vessel to reduce obstructions and maximise deck coverage. Additionally, it is crucial to examine the potential effects of induced boom vibrations on sensor data, as this could pose challenges when integrating sensors at the boom tip. Lastly, to ensure a reliable and practical measurement solution, the author recommends incorporating redundant LiDAR or RADAR sensors to maintain consistent and complete coverage of the supplier deck, even in the presence of obstructions.

References

[1] Huisman. “Unpublished internal company document”. In: (2022). URL: https://www.huismanequipment.com/en/ (visited on 03/04/2023).

[2] Hugo Díaz et al. “Market Needs, Opportunities and Barriers for the Floating Wind Industry”. In: *Journal of Marine Science and Engineering* 10.7 (2022). DOI: 10.3390/jmse10070934.

[3] Daniel Rippel et al. “A Review on the Planning Problem for the Installation of Offshore Wind Farms”. In: *IFAC-PapersOnLine* 52.13 (2019), pp. 1337–1342. DOI: https://doi.org/10.1016/j.ifacol.2019.11.384.

[4] *A Study of New Installation Concepts of Offshore Wind Farms by Means of Simulation Model*. International Ocean and Polar Engineering Conference. 2017, ISOPE-I-17–431. URL: https://onepetro.org/ISOPEIOPEC/proceedings-pdf/ISOPE17/All-ISOPE17/ISOPE-I-17-431/1256633/isope-i-17-431.pdf.

[5] *Simulative Comparison of Installation Concepts for Offshore Wind Farms*. Vol. All Days. International Ocean and Polar Engineering Conference. 2022. URL: https://onepetro.org/ISOPEIOPEC/proceedings-pdf/ISOPE22/All-ISOPE22/ISOPE-I-22-054/2788631/isope-i-22-054.pdf.

[6] U.S. Energy Information Administration. *International energy outlook 2021*. 2021. URL: https://www.eia.gov/outlooks/ieo/pdf/IEO2021_Narrative.pdf (visited on 06/19/2023).

[7] H. Díaz and C. Guedes Soares. “Review of the current status, technology and future trends of offshore wind farms”. In: *Ocean Engineering* 209 (2020), p. 107381. DOI: https://doi.org/10.1016/j.oceaneng.2020.107381.

[8] North of England P&I Association. *Ship to Ship Transfer*. 2019. URL: www.nepia.com/publications/ship-to-ship-transfer-briefing (visited on 06/14/2023).

[9] United States Government Accountability Office. *OFFSHORE WIND ENERGY*. 2020. URL: https://www.gao.gov/assets/gao-21-153-highlights.pdf (visited on 06/06/2023).

[10] Lake Manitoba Stewardship Board. *Annual Report 2007-08*. Lake Manitoba Stewardship Board, 2008.

[11] Sondre Sanden Tørdal. “Real-Time Motion Compensation in Ship-to-Ship Load Handling”. In: (2019). URL: https://uia.brage.unit.no/uia-xmlui/bitstream/handle/11250/2590151/thesis_no_articles.pdf?sequence=1.

[12] J. Ye. “Smart offshore heavy lift operations”. PhD thesis. 2020. DOI: 10.4233/uuid:469a5099-8241-45f7-b7fa-658497fa6142.

[13] MacGregor. *MacGregor Ship-to-Ship Compensation System*. 2017. URL: https://www.youtube.com/watch?v=5_QoSTrU2gc (visited on 06/14/2023).

[14] NOV. *Sealift Compensation System*. 2017. URL: https://www.youtube.com/watch?v=oJHl6g9R6z8 (visited on 06/14/2023).

[15] Ryan A. McKenzie and Rishad A. Irani. “Motion compensation for maritime cranes during time-varying operations at the pendulum’s natural frequency”. In: *Mechanism and Machine Theory* 168 (2022). DOI: 10.1016/j.mechmachtheory.2021.104573.

[16] Evert Lataire, Marc Vantorre, and Guillaume Delefortrie. “Captive model testing for ship to ship operations”. In: *MARSIM ’09 Conference, Proceedings*. 2009, p. 10. URL: http://hdl.handle.net/1854/LU-1060548.

[17] Evert Lataire et al. “Ship to ship interaction forces during lightering operations”. In: *2nd International conference on Ship Manoeuvring in Shallow and Confined Water: Ship to ship interaction*. 2011, pp. 211–222. URL: http://hdl.handle.net/1854/LU-1870267.

[18] OCIMF. *Mooring Load Analysis during Ship to Ship Tranfer Operations*. 2022. URL: https://www.ocimf.org/document-libary/669-sts-mooring-load-guide-2022-final-2/file (visited on 06/14/2023).

[19] Port of Gothenburg. *Ship to Ship Transfers*. 2016. URL: https://www.portofgothenburg.com/FileDownload/?contentReferenceID=1126 (visited on 06/14/2023).

[20] T. Fossen. *Handbook of Marine Craft Hydrodynamics and Motion Control*. John Wiley & Sons, Ltd, 2011. DOI: 10.1002/9781119994138.

[21] Jinwoo Kim et al. "Multi-sensor-based detection and tracking of moving objects for relative position estimation in autonomous driving conditions". In: *The Journal of Supercomputing* 76 (2020), pp. 8225–8247.

[22] Guoguang Du et al. "Vision-based robotic grasping from object localization, object pose estimation to grasp estimation for parallel grippers: a review". In: (2021). DOI: https://doi.org/10.1007/s10462-020-09888-5.

[23] Shuvo Kumar Paul et al. "Object detection and pose estimation from rgb and depth data for real-time, adaptive robotic grasping". In: 2021. DOI: https://doi.org/10.48550/arXiv.2101.07347.

[24] Jiaying Lin et al. "Environment Perception and Object Tracking for Autonomous Vehicles in a Harbor Scenario". In: *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*. 2020, pp. 1–6. DOI: 10.1109/ITSC45102.2020.9294618.

[25] Eulalia Balestrieri et al. "Sensors and Measurements for Unmanned Systems: An Overview". In: *Sensors* 21.4 (2021). DOI: 10.3390/s21041518.

[26] Ilham Faisal, Tito Purboyo, and Anton Ansori. "A Review of Accelerometer Sensor and Gyroscope Sensor in IMU Sensors on Motion Capture". In: *Journal of Engineering and Applied Sciences* 15 (2019), pp. 826–829. DOI: 10.36478/jeasci.2020.826.829.

[27] He Zhao and Zheyao Wang. "Motion Measurement Using Inertial Sensors, Ultrasonic Sensors, and Magnetometers With Extended Kalman Filter for Data Fusion". In: *IEEE Sensors Journal* 12.5 (2012), pp. 943–953. DOI: 10.1109/JSEN.2011.2166066.

[28] Kongsberg. *Title of the Website*. 2022. URL: https://www.kongsberg.com/contentassets/77678b37b4c94d51bbfd7d8abdf82c73/datasheet_mruh.pdf (visited on 01/17/2023).

[29] P. Zhang et al. "Navigation with IMU/GPS/digital compass with unscented Kalman filter". In: *IEEE International Conference Mechatronics and Automation, 2005*. Vol. 3. 2005, 1497–1502 Vol. 3. DOI: 10.1109/ICMA.2005.1626777.

[30] Jingang Yi et al. "IMU-based localization and slip estimation for skid-steered mobile robots". In: *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2007, pp. 2845–2850. DOI: 10.1109/IROS.2007.4399477.

[31] Ye Ding et al. "IMU-based iterative control for hip extension assistance with a soft exosuit". In: *2016 IEEE International Conference on Robotics and Automation (ICRA)*. 2016, pp. 3501–3508. DOI: 10.1109/ICRA.2016.7487530.

[32] Michal Kelemen et al. "Distance measurement via using of ultrasonic sensor". In: *Journal of Automation and Control* 3.3 (2015), pp. 71–74.

[33] Bernd Henning and Jens Rautenberg. "Process monitoring using ultrasonic sensor systems". In: *Ultrasonics* 44 (2006), e1395–e1399. DOI: https://doi.org/10.1016/j.ultras.2006.05.048.

[34] L. Koval, J. Vaňuš, and P. Bilík. "Distance Measuring by Ultrasonic Sensor". In: *IFAC-PapersOnLine* 49.25 (2016), pp. 153–158. DOI: https://doi.org/10.1016/j.ifacol.2016.12.026.

[35] Roni Stiawan et al. "An Ultrasonic Sensor System for Vehicle Detection Application". In: *Journal of Physics: Conference Series* 1204 (2019), p. 012017. DOI: 10.1088/1742-6596/1204/1/012017.

[36] Piero Zappi, Elisabetta Farella, and Luca Benini. "Tracking Motion Direction and Distance With Pyroelectric IR Sensors". In: *IEEE Sensors Journal* 10.9 (2010), pp. 1486–1494. DOI: 10.1109/JSEN.2009.2039792.

[37] Buddhika Amila. “Investigation of Infrared Sensors for Robot Navigation”. In: *IOSR Journal of Engineering* 09 (2019), pp. 11–17.

[38] T. Gandhi and M.M. Trivedi. “Pedestrian collision avoidance systems: a survey of computer vision based recent studies”. In: *2006 IEEE Intelligent Transportation Systems Conference*. 2006, pp. 976–981. DOI: 10.1109/ITSC.2006.1706871.

[39] Motaz Khader and Samir Cherian. “An introduction to automotive lidar”. In: *Texas Instruments* (2020).

[40] Ricardo Roriz, Jorge Cabral, and Tiago Gomes. “Automotive LiDAR Technology: A Survey”. In: *IEEE Transactions on Intelligent Transportation Systems* 23.7 (2022), pp. 6282–6297. DOI: 10.1109/TITS.2021.3086804.

[41] Jingyun Liu et al. “TOF Lidar Development in Autonomous Vehicle”. In: *2018 IEEE 3rd Optoelectronics Global Conference (OGC)*. 2018, pp. 185–190. DOI: 10.1109/OGC.2018.8529992.

[42] César Debeunne and D. Vivet. “A Review of Visual-LiDAR Fusion based Simultaneous Localization and Mapping”. In: *Sensors* 20 (2020), p. 2068. DOI: 10.3390/s20072068.

[43] John O. Woods and John A. Christian. “Lidar-based relative navigation with respect to noncooperative objects”. In: *Acta Astronautica* 126 (2016), pp. 298–311. DOI: https://doi.org/10.1016/j.actaastro.2016.05.007.

[44] Roberto Opromolla et al. “Uncooperative pose estimation with a LIDAR-based system”. In: *Acta Astronautica* 110 (2015), pp. 287–297. DOI: https://doi.org/10.1016/j.actaastro.2014.11.003. URL: https://www.sciencedirect.com/science/article/pii/S0094576514004354.

[45] Zhangjing Wang, Yu Wu, and Qingqing Niu. “Multi-Sensor Fusion in Automated Driving: A Survey”. In: *IEEE Access* 8 (2020), pp. 2847–2868. DOI: 10.1109/ACCESS.2019.2962554.

[46] Martin Pirkl and Franz-Jose Tospann. “HiVision millimeter-wave radar for enhanced vision systems in civil and military transport aircraft”. In: 1997.

[47] Taohua Zhou et al. “MMW Radar-Based Technologies in Autonomous Driving: A Review”. In: *Sensors* 20 (2020), p. 7283. DOI: 10.3390/s20247283.

[48] Xiao Wang et al. “On-Road Vehicle Detection and Tracking Using MMW Radar and Monovision Fusion”. In: *IEEE Transactions on Intelligent Transportation Systems* 17.7 (2016), pp. 2075–2084. DOI: 10.1109/TITS.2016.2533542.

[49] Abdul Sajeed Mohammed et al. “The Perception System of Intelligent Ground Vehicles in All Weather Conditions: A Systematic Literature Review”. In: *Sensors* 20 (2020). DOI: 10.3390/s20226532.

[50] “Unmanned surface vehicles: An overview of developments and challenges”. In: *Annual Reviews in Control* 41 (2016), pp. 71–93. DOI: https://doi.org/10.1016/j.arcontrol.2016.04.018.

[51] Yan Wang et al. “Pseudo-LiDAR From Visual Depth Estimation: Bridging the Gap in 3D Object Detection for Autonomous Driving”. In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019, pp. 8437–8445. DOI: 10.1109/CVPR.2019.00864.

[52] Nicholas Ayache and Francis Lustman. “Trinocular stereovision for robotics”. In: *Ieee transactions on pattern analysis and machine intelligence* (1991).

[53] S. Nedevschi et al. “3D lane detection system based on stereovision”. In: *Proceedings. The 7th International IEEE Conference on Intelligent Transportation Systems (IEEE Cat. No.04TH8749)*. 2004, pp. 161–166. DOI: 10.1109/ITSC.2004.1398890.

[54] InertialLabs. *Motion Reference Units MRU*. 2022. URL: https://inertiallabs.com/wp-content/uploads/2022/01/MRU_Datasheet.rev3_.3_Jan_2022-1.pdf (visited on 01/23/2023).

[55] SMC. *SMC IMU-10 Range*. 2022. URL: https://www.shipmotion.eu/files/IMU-10x%5C%20Motion%5C%20Sensor%5C%20datasheet.pdf (visited on 01/23/2023).

[56] Baumer. *Ultrasonic distance measuring sensors*. 2022. URL: https://www.baumer.com/medias/__secure__/UNAM_70U6131_S14_web_EN.pdf?mediaPK=8801191526430 (visited on 01/23/2023).

[57] Omron. *Ultrasonic Displacement Sensor E4PA-N*. 2022. URL: https://assets.omron.com/m/4e4bdd483c84da3a/original/D02E4PA0104-pdf.pdf (visited on 01/23/2023).

[58] Sharp. *GP2Y0A710K0F*. 2022. URL: https://global.sharp/products/device/lineup/data/pdf/datasheet/gp2y0a710k_e.pdf (visited on 01/23/2023).

[59] Sharp. *GP2Y0D02YK0F*. 2022. URL: https://global.sharp/products/device/lineup/data/pdf/datasheet/gp2y0d02yk_e.pdf (visited on 01/23/2023).

[60] Aeva. *Aeries II*. 2023. URL: https://www.aeva.com/aeries-ii/ (visited on 01/25/2023).

[61] Baraja. *Spectrum HD25*. 2023. URL: https://www.baraja.com/en/spectrum-hd25 (visited on 01/24/2023).

[62] Cepton. *Vista-X Family*. 2022. URL: https://www.cepton.com/products/vista-x (visited on 01/24/2023).

[63] Hesai. *AT128*. 2023. URL: https://www.hesaitech.com/en/AT128 (visited on 01/24/2023).

[64] Innoviz. *InnovizTwo*. 2023. URL: https://innoviz.tech/innoviztwo (visited on 01/24/2023).

[65] Luminar. *A platform for safety and autonomy. Built from the chip-level up*. 2023. URL: https://www.luminartech.com/technology#iris (visited on 01/24/2023).

[66] Ouster. *OS2*. 2022. URL: https://ouster.com/products/scanning-lidar/os2-sensor/ (visited on 01/24/2023).

[67] Robosense. *RS-LiDAR-M1*. 2022. URL: https://www.robosense.ai/en/rslidar/RS-LiDAR-M1 (visited on 01/24/2023).

[68] Velodyne. *Alpha Prime*. 2022. URL: https://velodynelidar.com/products/alpha-prime/ (visited on 01/23/2023).

[69] Zvision. *ML-Xs*. 2022. URL: http://www.zvision.xyz/en/h-col-279.html (visited on 01/23/2023).

[70] Continental. *Front radar Sensor*. 2022. URL: https://www.bosch-mobility-solutions.com/en/solutions/sensors/front-radar-sensor/ (visited on 01/24/2023).

[71] Continental. *ARS540*. 2023. URL: https://www.continental-automotive.com/en-gl/Passenger-Cars/Autonomous-Mobility/Enablers/Radars/Long-Range-Radar/ARS540 (visited on 01/24/2023).

[72] Continental. *ARS441*. 2023. URL: https://www.continental-automotive.com/en-gl/Passenger-Cars/Autonomous-Mobility/Enablers/Radars/Long-Range-Radar/ARS441 (visited on 01/24/2023).

[73] Intel. *Intel RealSense Depth Camera D435*. 2023. URL: https://www.intelrealsense.com/depth-camera-d435/ (visited on 01/25/2023).

[74] Orbbec. *Astra Series*. 2023. URL: https://orbbec3d.com/index/Product/info.html?cate=38&id=36 (visited on 01/25/2023).

[75] StereoLabs. *Meet ZED 2*. 2023. URL: https://www.stereolabs.com/zed-2/ (visited on 01/25/2023).

[76] Sondre Sanden Tørdal, Per-Ove Løvsland, and Geir Hovland. “Testing of wireless sensor performance in Vessel-to-Vessel Motion Compensation”. In: *IECON 2016 - 42nd Annual Conference of the IEEE Industrial Electronics Society*. 2016, pp. 654–659. DOI: 10.1109/IECON.2016.7793951.

[77] Sondre Sanden Tørdal and Geir Hovland. “Relative Vessel Motion Tracking using Sensor Fusion, Aruco Markers, and MRU Sensors”. In: *Modeling, Identification and Control* 38.2 (2017), pp. 79–93. DOI: doi:10.4173/mic.2017.2.3.

[78] Sondre Sanden Tørdal, Witold Pawlus, and Geir Hovland. “Real-time 6-DOF Vessel-to-Vessel Motion Compensation using laser tracker”. In: vol. 2017-October. Cited by: 4. 2017, pp. 1–9. DOI: 10.1109/OCEANSE.2017.8084756.

[79] Sondre Sanden Tørdal and Geir Hovland. “Ship-to-Ship State Observer Using Sensor Fusion and the Extended Kalman Filter”. In: *Journal of Offshore Mechanics and Arctic Engineering* 141.4 (2019). DOI: 10.1115/1.4041643. URL: https://doi.org/10.1115/1.4041643.

[80] Hans K.R. Holen, Alexander M. Sjøberg, and Olav Egeland. “Estimation of Ship-Deck Motion using LIDAR,Gyroscopes and Cameras”. In: *Modeling, Identification and Control* 42.3 (2021), pp. 99–112. DOI: 10.4173/mic.2021.3.1.

[81] Zhuolin Wang and Yingjun Zhang. “Estimation of ship berthing parameters based on Multi-LiDAR and MMW radar data fusion”. In: *Ocean Engineering* 266 (2022), p. 113155. DOI: https://doi.org/10.1016/j.oceaneng.2022.113155.

[82] Yougang Sun et al. “Experimental design and development of heave compensation system for marine crane”. In: *Mathematical modelling of engineering problems* 1.1 (2014), pp. 15–20. URL: http://dx.doi.org/10.18280/mmep.010204.

[83] Bilgin Bozkurt, Melek Ertogan, and Şebnem Helvacıoğlu. “2-D Control of Payload on an Offshore Crane during Ship-to-ship Load Transfer Operation”. In: *2022 10th International Conference on Control, Mechatronics and Automation (ICCMA)*. 2022, pp. 142–147. DOI: 10.1109/ICCMA56665.2022.10011458.

[84] Johan Lindal Haug. “Development of a vision-based measurement system for relative motion compensation”. MA thesis. Universitetet i Agder; University of Agder, 2014.

[85] Xian-Feng Han et al. “A review of algorithms for filtering the 3D point cloud”. In: *Signal Processing: Image Communication* 57 (2017), pp. 103–112. DOI: https://doi.org/10.1016/j.image.2017.05.009.

[86] Matteo Munaro et al. “A Software Architecture for RGB-D People Tracking Based on ROS Framework for a Mobile Robot”. In: vol. 466. 2013, pp. 53–68. DOI: 10.1007/978-3-642-35485-4_5.

[87] Julie Digne and Carlo de Franchis. “The Bilateral Filter for Point Clouds”. In: *Image Processing On Line* 7 (2017), pp. 278–287. DOI: https://doi.org/10.5201/ipol.2017.179.

[88] Marc Alexa et al. “Point set surfaces”. In: *Proceedings Visualization, 2001. VIS ’01.* (2001), pp. 21–29, 537.

[89] Yujing Sun, Scott Schaefer, and Wenping Wang. “Denoising point sets via L0 minimization”. In: *Computer Aided Geometric Design* 35-36 (2015), pp. 2–15. DOI: https://doi.org/10.1016/j.cagd.2015.03.011.

[90] E. Grilli, F. Menna, and F. Remondino. “A REVIEW OF POINT CLOUDS SEGMENTATION AND CLASSIFICATION ALGORITHMS”. In: *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* XLII-2/W3 (2017), pp. 339–344. DOI: 10.5194/isprs-archives-XLII-2-W3-339-2017.

[91] PCL. *Point Cloud Library*. 2023. URL: http://pointclouds.org (visited on 03/08/2023).

[92] Ruwen Schnabel, Patrick Degener, and Reinhard Klein. “Completion and Reconstruction with Primitive Shapes”. In: *Comput. Graph. Forum* 28 (2009), pp. 503–512. DOI: 10.1111/j.1467-8659.2009.01389.x.

[93] MathWorks. *k-means clustering - MATLAB*. 2023. URL: https://mathworks.com/help/stats/kmeans.html (visited on 03/08/2023).

[94] Vladeta Stojanovic. “Digital Twins for Indoor Built Environments”. PhD thesis. 2021. DOI: 10.25932/publishup-50913.

[95] MathWorks. *Point Cloud Processing - MATLAB*. 2023. URL: https://nl.mathworks.com/products/matlab.html.

[96] Epic Games. *Unreal Engine*. 2023. URL: https://www.unrealengine.com/en-US/?state=%5C%2F5.2%5C%2Fen-US%5C%2F (visited on 06/25/2023).

[97] Open Robotics. *Gazebo*. 2023. URL: https://gazebosim.org/home (visited on 06/25/2023).

[98] MathWorks. *Simulink 3D Animation - MATLAB*. 2023. URL: https://nl.mathworks.com/products/3d-animation.html.

[99] MathWorks. *AUV Deep Dive - MATLAB & Simulink*. 2023. URL: https://nl.mathworks.com/videos/series/auv-deep-dive.html (visited on 06/23/2023).

[100] Jan De Nul. *Les Alizés*. 2023. URL: `https://www.jandenul.com/sites/default/files/2020-06/Les%5C%20Aliz%5C%C3%5C%A9s%5C%20%5C%28EN%5C%29_.pdf` (visited on 06/25/2023).

[101] José María Martínez-Otzeta et al. "RANSAC for Robotic Applications: A Survey". In: *Sensors* 23.1 (2023). ISSN: 1424-8220. DOI: `10.3390/s23010327`.

[102] G. Peters and J. H. Wilkinson. "The least squares problem and pseudo-inverses". In: *The Computer Journal* 13.3 (Jan. 1970), pp. 309–316. DOI: `10.1093/comjnl/13.3.309`.

[103] S. Y. Chen. "Kalman Filter for Robot Vision: A Survey". In: *IEEE Transactions on Industrial Electronics* 59.11 (2012), pp. 4409–4420. DOI: `10.1109/TIE.2011.2162714`.

[104] Emiel Por, Maaike van Kooten, and Vanja Sarkovic. "Nyquist-Shannon sampling theorem". In: *Leiden University* 1.1 (2019).

A

Scientific Research Paper

On the following pages, a summary of the thesis is presented in the form of a scientific research paper.

Relative Heave Measurement During Ship-to-Ship (STS) Transfer of Cargo

1st P. Correia [*], 2ndDr. ir. V. Reppa [†] , 3rd Dr. ir. Y. Pang [†] , 4th ir. J. Dijkstra [‡]

[*] *MSc. Multi-Machine Engineering - Delft University of Technology*
[†] *Faculty of 3mE — Marine and Transport Technology*
[‡] *Huisman Equipment B.V.*

***Abstract*—Ship-to-Ship (STS) cargo transfers can significantly improve the efficiency of offshore installations while reducing their associated costs. However, the added complexity of cargo transfers involving ship-mounted cranes at sea poses significant challenges. To address this, crane manufacturers have introduced Relative Heave Compensation (RHC) systems to assist crane operators. Nevertheless, concerns have emerged among installation vessel companies regarding the relative heave measurement systems, which rely on Motion Reference Unit (MRU) sensors placed on both ships. Given that many supplier vessels lack the required sensors and the communication protocols pose bureaucratic hurdles, this research focuses on proposing novel, feeder-independent relative heave measuring solutions. This work introduces two unexplored sensor units, namely the LiDAR-MRU and RADAR-MRU. In the absence of actual sensor data, a simulation workflow using Simulink and Unreal Engine (UE) is presented to generate synthetic data. After, four distinct measurement solutions are developed, implemented, and evaluated using the simulated data in MATLAB. The implemented solutions estimate relative heave distance and speed with accuracy, ranging from 0.3-5.2% and 0.8-4.3% of the reference's maximum amplitude, and processing times from 1.8-$91.4[ms]$. The results underscore their potential for field implementation. However, future validation is needed with actual sensor data.**

***Index Terms*—Ship-to-Ship, STS, Relative Heave, Sensing, Measurement, Data Processing and Analysis, Point Cloud**

I. Introduction

A. Research Background

The global increase in energy consumption [1] and the need for more sustainable energy solutions have driven the growth of the offshore wind sector [2]. As the number of wind farm installations proliferates, new projects extend farther from the coast into deeper waters. Consequently, Wind Turbine Installation Vessels (WTIV) now have to cover greater distances, leading to increased installation downtime and higher economic costs. To mitigate these challenges, the offshore-feeder concept [3] has emerged as a promising solution to streamline operations, reduce installation costs and circumvent regulatory constraints, like the Jones Act [4]. Instead of placing the entire responsibility for transporting and installing components on the WTIVs, this approach involves maintaining installation vessels at sea while employing cheaper feeder vessels for a continuous supply of components.

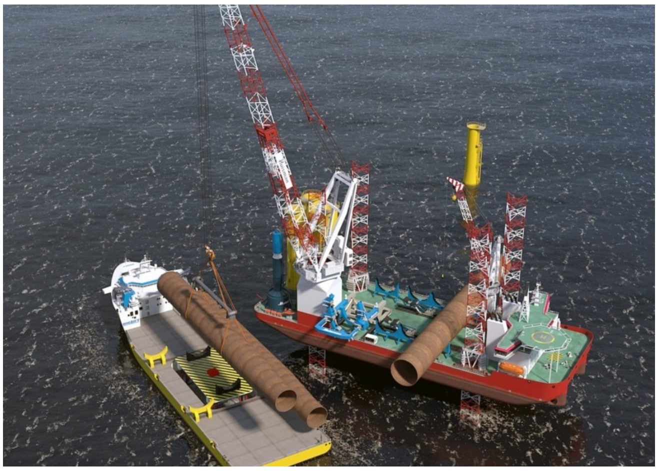


Fig. 1: Ship-to-Ship cargo transfers between two seagoing vessels.

Nevertheless, Ship-to-Ship (STS) cargo transfers, depicted in Figure 1, are inherently hazardous and complex due to the unpredictable nature of the sea [5]. Recognising this, maritime crane companies offer Relative Heave Compensation (RHC), an advancement of Active Heave Compensation (AHC) systems, to assist crane operators during the transferring procedure. These systems mitigate the impact of heave motion between ships, known as relative heave, thereby reducing the risk of load collisions with the ship's deck and simplifying the load connection process.

The existing state-of-practice RHC systems offered by prominent companies, namely Huisman [6], NOV [7] and MacGregor [8], employ one Motion Reference Unit (MRU) on each vessel and a wireless communication link to measure relative heave. However, concerns have been raised by crane vessel owners regarding the dependence on sensors located on feeder vessels, citing issues such as the unavailability of MRU sensors on feeder vessels and the hurdles associated with sharing data among vessels owned by different companies. Therefore, this study aims to address these concerns by proposing feeder-independent measuring systems.

B. Literature Review

Although RHC systems have gained considerable attention in the offshore industry, academic research is still limited. Only two authors [9], [10] have directly addressed the load-transferring problem. McKenzie and Irani [9] suggested future

This research was supported by Huisman Equipment B.V.

research on a robust sensor unit is critical to track the relative motions of the supplier vessel. Notably, Tørdal [10] has dedicated substantial efforts to address this specific issue, publishing a series of papers [11]–[14] that explore various sensor configurations to measure these relative motions.

Tørdal introduced measurement systems, employing sensors such as cameras, MRUs, and Laser Trackers. However, all proposed systems relied on the sensors installed on each ship. Tørdal primarily focused on augmenting the state-of-practice systems with a third sensor to measure the absolute distance between ships rather than developing a feeder-independent solution. Nevertheless, he suggested that integrating alternative, more robust sensors, such as time-of-flight and vision-based sensors, could potentially obviate the need for MRUs installed on the feeder vessels.

Remarkably, Holen, Sjøberg, and Egeland [15] proposed the only feeder-independent measuring solution, combining vision and LiDAR sensors for relative motion estimation. Although their experimental results were promising, using camera sensors for offshore applications has its inherent limitations, namely reduced depth resolution and vulnerability to adverse weather conditions.

The literature overview in Table I reveals a lack of solutions for measuring ship-to-ship relative motions, especially with feeder-independent sensor configurations. With Laser trackers and cameras being unsuitable for outdoor and offshore applications, 3D time-of-flight sensors rise as a promising research direction.

TABLE I: Overview of the state-of-the-art measuring solutions for ship-to-ship relative motions.

Ref.	Crane Vessel				Supply Vessel	
	MRU	Camera	Laser Tracker	LiDAR	MRU	Marker*
[11]	✓	✗	✗	✗	✓	✗
[12]	✓	✓	✗	✗	✓	✓
[13]	✗	✗	✓	✗	✗	✓
[14]	✓	✗	✓	✗	✓	✓
[15]	✓	✓	✗	✓	✗	✓

Note: (✓) used; (✗) not used; *visual reference point

Based on Balestrieri et al. [16] extensive study on available sensors for unmanned systems, LiDAR and RADAR were found to be suitable 3D time-of-flight sensors due to their accuracy and robustness for the intended application. Given the necessity to measure both the crane vessel's motion and the relative movements of the feeder vessel, the sensor units LiDAR-MRU, RADAR-MRU, or LiDAR-RADAR-MRU emerged as viable candidates.

C. Problem Definition

STS cargo transfers require a robust feeder-independent relative heave measurement system. However, the literature review reveals a lack of research on this topic. Therefore, this research aims to **propose, implement and compare novel methods for measuring relative heave distance and speed in offshore ship-to-ship operations, relying exclusively on sensors installed on the crane vessel**.

Due to this study's time constraints, not all possible sensor configurations can be explored. Therefore, this study solely focuses on two novel sensor configurations, the LiDAR-MRU and RADAR-MRU, which have not been previously investigated in the existing literature. The proposed sensor setup entails the installation of both sensors on the crane vessel, with the RADAR or LiDAR sensors strategically positioned at the boom tip to ensure direct coverage of the feeder vessel.

D. Outline

The paper is organised as follows: Section II-A develops a mathematical model to describe relative heave in STS operations, offering a comprehensive understanding of the problem. In the absence of actual sensor data, Section II-B implements a simulation environment to simulate STS operations using pre-implemented RADAR and LiDAR models. In Section II-C, distinct measuring methods and their corresponding implementations are introduced. These implementations convert the simulated sensor data into relative heave estimates through data processing and analysis algorithms supported by a Kalman Filter implemented in Section II-D. Finally, Section III analyses the performance of the implemented measuring solutions, and Section IV concludes the paper, suggesting future research avenues.

II. Methods

A. Relative Heave Modelling

Before embarking on the modelling of the problem, the simplifications made can be summarised as follows:

1) The crane and supplier vessels are modelled as independent rigid bodies.
2) The swinging motions of the load are neglected.
3) The wire rope is assumed to be perfectly rigid and without any elasticity.
4) The kinematics of the crane are not taken into account.

It is important to remark that these assumptions align with current industry practices.

Figure 2 describes the kinematics of the STS cargo transfer problem, and Table II describes the annotations used. The vector notation and kinematic formulations used were based on the work introduced by Fossen [17] and used by Tørdal [10] to model STS operations.

TABLE II: Description of the kinematic annotations presented in Figure 2

Symbol	Description
$\{b_1\}$	Coordinate system of ship 1.
$\{b_2\}$	Coordinate system of ship 2.
$\{g\}$	Global coordinate system.
$\{m\}$	Motion Reference Unit (MRU) mounted on ship body 1.
$\{s\}$	Exteroceptive sensor mounted on boom tip of ship body 1.
D	Distance between points $\boldsymbol{p_t}$ and $\boldsymbol{p_d}$
$\boldsymbol{p_t}$	The crane tip point where the Wire Exit Point (WEP) is located
$\boldsymbol{n_w}$	The unit vector following the wire connecting the crane and the hook.
$\boldsymbol{p_d}$	The point where a line in the direction of $\boldsymbol{n_w}$ intersects the ship 2 deck.

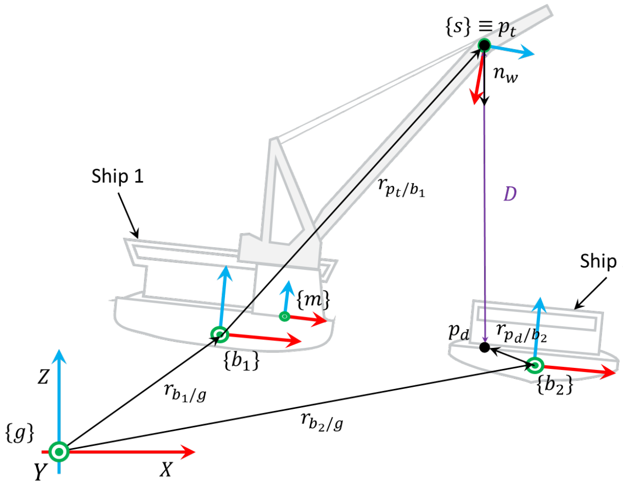


Fig. 2: Kinematics describing relative heave in STS cargo transfer operations.

From Figure 2, the relative heave distance (D) corresponds to the distance between $\boldsymbol{p_t}$ and $\boldsymbol{p_d}$:

$$\boldsymbol{r}^g_{p_t/g} + \boldsymbol{n^g_w} \cdot D = \boldsymbol{r}^g_{p_d/g} \tag{1}$$

Where $\boldsymbol{r}^g_{p/g}$ denotes the coordinates of a generic point ($\boldsymbol{p}$) in the global coordinate system $\{g\}$, in this case:

$$\boldsymbol{r}^g_{p_t/g} = [x^g_{p_t}, y^g_{p_t}, z^g_{p_t}]^T, \quad \boldsymbol{r}^g_{p_d/g} = [x^g_{p_d}, y^g_{p_d}, z^g_{p_d}]^T \tag{2}$$

Assuming no load swinging motions occur, $\boldsymbol{n^g_w}$ remains aligned with gravity and can be described as follows:

$$\boldsymbol{n^g_w} = \begin{bmatrix} 0 \\ 0 \\ -1 \end{bmatrix} \tag{3}$$

Combining Equations (1) and (3):

$$\boldsymbol{r}^g_{p_t/g} - \boldsymbol{r}^g_{p_d/g} = \begin{bmatrix} 0 \\ 0 \\ D \end{bmatrix} \tag{4}$$

Rewriting Equation (4) using $\{b_1\}$ and $\{b_2\}$ body frames:

$$\boldsymbol{r}^g_{b_1/g} + \boldsymbol{R}^g_{b_1}(\boldsymbol{\Theta}_{gb_1})\boldsymbol{r}^{b_1}_{p_t/b_1} - \boldsymbol{r}^g_{b_2/g} - \boldsymbol{R}^g_{b_2}(\boldsymbol{\Theta}_{gb_2})\boldsymbol{r}^{b_2}_{p_d/b_2} = \begin{bmatrix} 0 \\ 0 \\ D \end{bmatrix} \tag{5}$$

Here, $\boldsymbol{r}$ denotes the ships' position, and $\boldsymbol{\Theta}$ represents their attitude relative to $\{g\}$.

$$\boldsymbol{r}^g_{b_1/g} = \begin{bmatrix} x_{b_1} \\ y_{b_1} \\ z_{b_1} \end{bmatrix}, \quad \boldsymbol{\Theta}_{gb_1} = \begin{bmatrix} \phi_{b_1} \\ \theta_{b_1} \\ \psi_{b_1} \end{bmatrix} \tag{6}$$

$$\boldsymbol{r}^g_{b_2/g} = \begin{bmatrix} x_{b_2} \\ y_{b_2} \\ z_{b_2} \end{bmatrix}, \quad \boldsymbol{\Theta}_{gb_2} = \begin{bmatrix} \phi_{b_2} \\ \theta_{b_2} \\ \psi_{b_2} \end{bmatrix} \tag{7}$$

Additionally, $\boldsymbol{R}^g_b$ is the rotation rotation matrix from frame $\{b\}$ to $\{g\}$, commonly represented as a combination of rotations around the Euler angles $\boldsymbol{\Theta}gb = [\phi, \theta, \psi]^T$:

$$\boldsymbol{R}^g_b(\boldsymbol{\Theta}_{gb}) = \boldsymbol{R}_z(\psi)\boldsymbol{R}_y(\theta)\boldsymbol{R}_x(\phi) \tag{8}$$

Where $\boldsymbol{R}_x$, $\boldsymbol{R}_y$ and $\boldsymbol{R}_z$ are the well established rotation matrices used around x-, y- and z-axis. Considering the coordinate systems used are left-handed and rotations are extrinsic, Equation (8) expands to:

$$\boldsymbol{R}^g_b(\boldsymbol{\Theta}_{gb}) = \begin{bmatrix} c_\psi c_\theta & c_\phi s_\psi + c_\psi s_\phi s_\theta & s_\phi s_\psi - c_\phi c_\psi s_\theta \\ -c_\theta s_\psi & c_\phi c_\psi - s_\phi s_\psi s_\theta & c_\psi s_\phi + c_\phi s_\psi s_\theta \\ s_\theta & -c_\theta s_\phi & c_\phi c_\theta \end{bmatrix} \tag{9}$$

The $\boldsymbol{p_t}$ position relative to $\{b_1\}$ is assumed static, resulting in:

$$\boldsymbol{r}^{b_1}_{p_t/b_1} = \begin{bmatrix} x_{p_t} \\ y_{p_t} \\ z_{p_t} \end{bmatrix}, \quad \dot{\boldsymbol{r}}^{b_1}_{p_t/b_1} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \tag{10}$$

Lastly, the unknown $\boldsymbol{p_d}$ position changes over time, but ship deck elevation is constant, therefore:

$$\boldsymbol{r}^{b_2}_{p_d/b_2} = \begin{bmatrix} x_{p_d} \\ y_{p_d} \\ z_{p_d} \end{bmatrix}, \quad \dot{\boldsymbol{r}}^{b_2}_{p_d/b_2} = \begin{bmatrix} \dot{x}_{p_d} \\ \dot{y}_{p_d} \\ 0 \end{bmatrix} \tag{11}$$

Finally, solving the system of Equations (5) for D along with the unknowns x_{p_d} and y_{p_d} mentioned in equation (11), the relative heave distance can be computed by f:

$$D = f(\boldsymbol{r}^g_{b_1}, \boldsymbol{\Theta}_{gb_1}, \boldsymbol{r}^g_{b_2}, \boldsymbol{\Theta}_{gb_2}, \boldsymbol{r}^{b_1}_{p_t}, z_{p_d}) \ , \quad z_{p_d} = const. \tag{12}$$

Furthermore, taking the time derivative of eq. (5), the following expression is obtained:

$$\dot{\boldsymbol{r}}^g_{b_1/g} + \dot{\boldsymbol{R}}^g_{b_1}\boldsymbol{r}^{b_1}_{p_t/b_1} - \dot{\boldsymbol{r}}^g_{b_2/g} - (\dot{\boldsymbol{R}}^g_{b_2}\boldsymbol{r}^{b_2}_{p_d/b_2} + \boldsymbol{R}^g_{b_2}\dot{\boldsymbol{r}}^{b_2}_{p_d/b_2}) = \begin{bmatrix} 0 \\ 0 \\ \dot{D} \end{bmatrix} \tag{13}$$

Solving the system of Equations (13) for $\dot{D}$ along with the unknowns $\dot{x}_{p_d}$ and $\dot{y}_{p_d}$, the relative heave speed can be computed by g:

$$\dot{D} = g(\dot{\boldsymbol{r}}^g_{b_1}, \boldsymbol{w}_{gb_1}, \boldsymbol{\Theta}_{gb_1}, \dot{\boldsymbol{r}}^g_{b_2}, \boldsymbol{w}_{gb_2}, \boldsymbol{\Theta}_{gb_1}, z_{p_d}) \tag{14}$$

The functions f and g have been excluded due to their substantial size.

B. *Ship-to-Ship Simulation*

Acquiring sensor data is a fundamental step before exploring various data processing and analysis methods. In the absence of actual sensor data, this thesis establishes a co-simulation framework to generate synthetic data. An analogous example provided by MathWorks [18] inspired this framework. The structure, depicted in Figure 3, outlines the flow from data acquisition to result extraction. The simulation comprises two concurrent cycles: (a) deriving relative heave ground truth values (using Equations (12) and (14)) and (b) collecting 3D Point Cloud data at each time step.

The Simulink model establishes the ships' motions and applies the in-built RADAR and LiDAR models to generate sensor data. Additionally, the crane vessel attitude is stored, assuming an ideal MRU is available. The UE implementation entails the 3D environment containing the ships and crane models. These models move according to the motions generated in Simulink. Figure 4 depicts four subsequent time frames

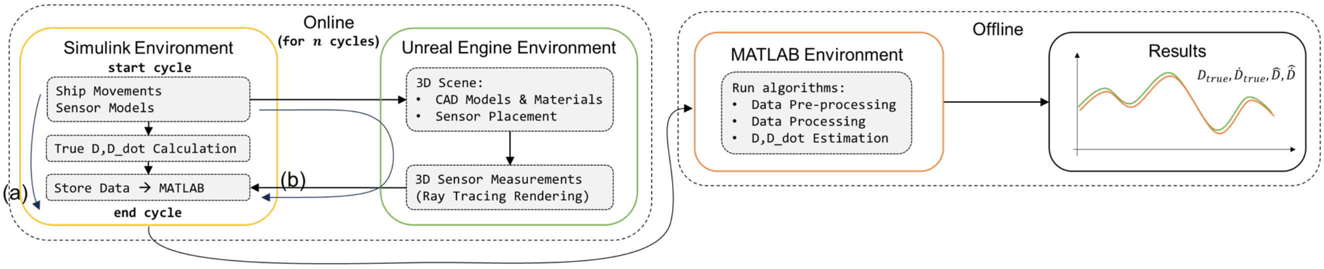
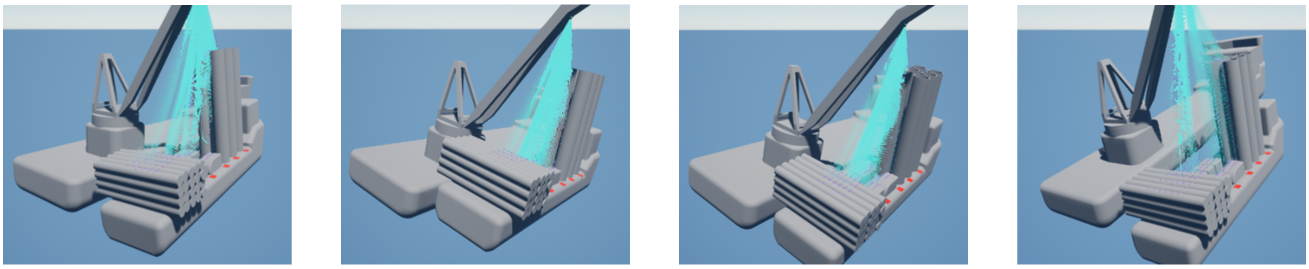


Fig. 3: Overview of the data generation and processing workflow using Simulink, Unreal Engine (UE), and MATLAB.



(a) Time Step n_1

(b) Time Step n_2

(c) Time Step n_3

(d) Time Step n_4

Fig. 4: Progression of ships position and orientation over time in Unreal Engine (UE) with $n_1 < n_2 < n_3 < n_4$

of the UE simulation display window. The sensor rays are depicted in blue and in red highly reflective markers.

The simulation data is stored in a time-series MATLAB format, including RADAR and LiDAR point clouds, MRU measurements, and true values for relative heave at each time step over the simulation window. Figure 5 showcases an exemplar LiDAR point cloud obtained at a specific time step. A similar, less dense RADAR point cloud is also acquired but is not presented here.

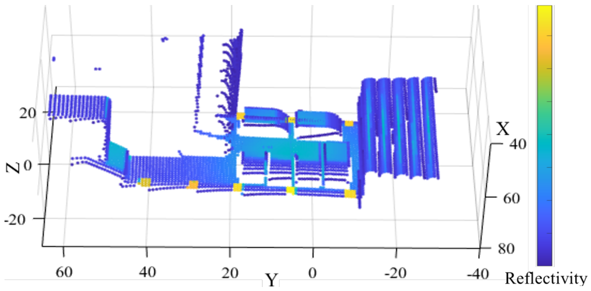


Fig. 5: LiDAR point cloud oblique view at time step t, where the colour scale indicates reflectivity.

C. Relative Heave Measurement

The first step of the measuring solutions scheme, depicted in Figure 6, is to convert the sensor data into relative heave measurements. This step, called the Measuring Method, consists of a series of Data Processing and Analysis steps. In this work, three distinct methods have been developed: *deck detection* (implemented for both RADAR-MRU and LiDAR MRU sensor units), *reflectors detection* (exclusive to LiDAR-MRU), and *Doppler-based* (exclusive to RADAR-MRU).

1) Deck Detection Method: This method relies on the spatial coordinates of all captured points to detect the ship deck and determine the relative heave distance (D). As shown in Figure 7, the sensor records N points (x, y, z) in frame $\{s\}$. These points are used to establish a best-fitting plane equation, employing a RANSAC algorithm [19]:

$$ax + by + cz + d = 0 \tag{15}$$

Furthermore, using the MRU attitude measurements, the orientation vector $\boldsymbol{n_w}$ relative to $\{s\}$ can be computed as:

$$\hat{\boldsymbol{n}}_w^s = \boldsymbol{R}_{b_1}^s \boldsymbol{R}_g^{b_1} \hat{\boldsymbol{n}}_w^g \tag{16}$$

Combining Equation (15) and (16), the line-plane intersection point can be determined and, with a few mathematical manipulations D, can be computed as:

$$D = \frac{-d}{a\hat{n}_{w_x}^s + b\hat{n}_{w_y}^s + c\hat{n}_{w_z}^s} \tag{17}$$

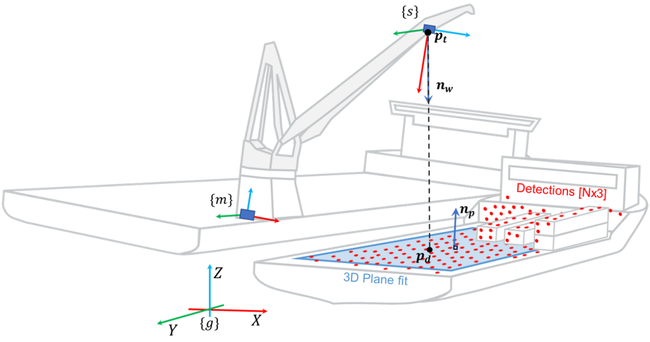


Fig. 7: Schematic of the *deck detection* method.

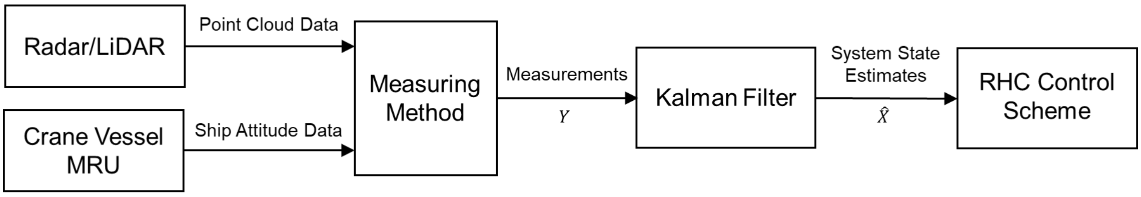


Fig. 6: Relative heave measurement system scheme, where: $Y = D$ or $Y = \dot{D}$ and $\hat{X} = [\hat{D} \quad \hat{\dot{D}}]$

2) Reflectors Detection Method: This method closely resembles *deck detection*, with a key difference: it exclusively utilises points corresponding to retro-reflectors positioned on the supplier vessel, as illustrated in Figure 8. The selection of these points is determined based on the LiDAR's intensity readings:

$$\texttt{Exclude} \quad if \quad \texttt{Intensity} \leq \texttt{Threshold} \tag{18}$$

After excluding all other points, the method detects the plane equation and computes D. This slight modification is expected to enhance both computational efficiency and robustness.

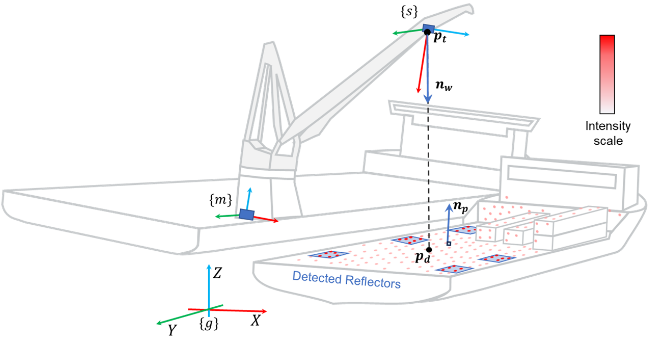


Fig. 8: Schematic of the reflector detection method.

3) Doppler-Based Method: This method leverages the RADAR's Doppler velocity measurements to determine the supplier vessel's relative velocity and compute its relative heave speed ($\dot{D}$).

Based on Figure 9, the position of point $\boldsymbol{p}$ can be represented as the sum of two vectors:

$$\boldsymbol{r}^s_{p/s} = \boldsymbol{r}^s_{b/s} + \boldsymbol{R}^s_b \boldsymbol{r}^b_{p/b} \tag{19}$$

One describes the position of the body reference frame $\{b\}$ in the sensor frame $\{s\}$, and the other describes the position of point $\boldsymbol{p}$ in $\{b\}$.

The derivative of Equation (19) can be expressed as:

$$\dot{\boldsymbol{r}}^s_{p} = \dot{\boldsymbol{r}}^s_{b/s} + \dot{\boldsymbol{R}}^s_b \boldsymbol{r}^b_{p/b} \tag{20}$$

As found in this study, introducing rotational motions would require tracking the body-fixed reference, which is not feasible with Doppler velocity measurements alone. Therefore, rotations are assumed to be negligible in this method. Hence,

$$\dot{\boldsymbol{r}}^s_{p/s} = \dot{\boldsymbol{r}}^s_{b/s} \tag{21}$$

Note, $\dot{\boldsymbol{r}}^s_{b/s}$ denotes the feeder vessel's linear velocity relative to the sensor:

$$\dot{\boldsymbol{r}}^s_{b/s} = \begin{bmatrix} \dot{x}_b \\ \dot{y}_b \\ \dot{z}_b \end{bmatrix} \tag{22}$$

For each RADAR detection, the azimuth (β) and elevation (α) angle of the reflected beam direction ($\boldsymbol{n}_b$) are captured. Therefore, $\hat{\boldsymbol{n}}^s_b$ can be computed by:

$$\hat{\boldsymbol{n}}^s_b = \begin{bmatrix} cos(\alpha)cos(\beta) \\ -cos(\alpha)sin(\beta) \\ sin(\alpha) \end{bmatrix} \tag{23}$$

Combining Equations (21), (22) and (23):

$$v_m = \dot{x}_b cos(\alpha)cos(\beta) - \dot{y}_b cos(\alpha)sin(\beta) + \dot{z}_b sin(\alpha) \tag{24}$$

As the RADAR will measure several v_m, the feeder's relative velocity can be determined by solving the following system of equations for n collected data points:

$$\underbrace{\begin{bmatrix} c_{\alpha_1}c_{\beta_1} & c_{\alpha_1}s_{\beta_1} & s_{\alpha_1} \\ c_{\alpha_2}c_{\beta_2} & c_{\alpha_2}s_{\beta_2} & s_{\alpha_2} \\ \vdots & \vdots & \vdots \\ c_{\alpha_n}c_{\beta_n} & c_{\alpha_n}s_{\beta_n} & s_{\alpha_n} \end{bmatrix}}_{C} \begin{bmatrix} \dot{x}_b \\ \dot{y}_b \\ \dot{z}_b \end{bmatrix} = \underbrace{\begin{bmatrix} v_{m1} \\ v_{m2} \\ v_{m3} \\ \vdots \\ v_{mn} \end{bmatrix}}_{V_m} \tag{25}$$

The least squares solution for the previous system can be obtained by utilising the pseudo-inverse of matrix C:

$$\dot{\boldsymbol{r}}^s_{b/s} = C^+ \cdot V^s_m \tag{26}$$

The last step to obtain the approximate relative heave speed ($\dot{D}$) is to project the relative velocity vector of the ship $\dot{\boldsymbol{r}}^s_{b/s}$ in the heave direction (Equation (16)).

$$\dot{D} = \dot{\boldsymbol{r}}^s_{b/s} \cdot n^s_w \tag{27}$$

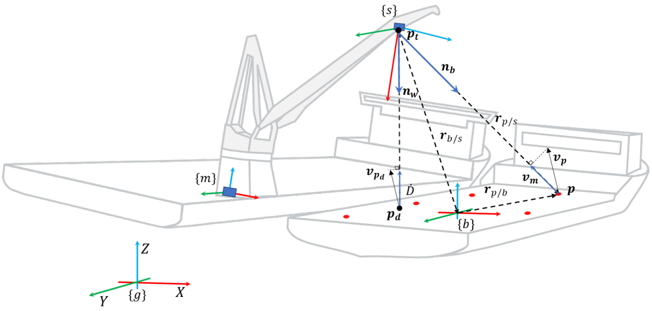


Fig. 9: Schematic of the *Doppler-based* method derivation.

D. Relative Heave Estimation

For RHC systems, noise-free real-time measurements are essential. Therefore, a Kalman Filter (KF) is implemented for this application [20].

1) Process Model: Given the considerable size of both ships, the relative heave acceleration is assumed to be constant. This assumption allows the formulation of the discrete-time state space model as:

$$\boldsymbol{x_s}[k+1] = A\boldsymbol{x_s}[k] + G\boldsymbol{w} \tag{28}$$

$$A = \begin{bmatrix} 1 & T_s & \frac{1}{2}(T_s)^2 \\ 0 & 1 & T_s \\ 0 & 0 & 1 \end{bmatrix}, \ G = \begin{bmatrix} \frac{1}{2}(T_s)^2 \\ T_s \\ 1 \end{bmatrix}, \ \boldsymbol{x_s}[k] = \begin{bmatrix} D[k] \\ \dot{D}[k] \\ \ddot{D}[k] \end{bmatrix} \tag{29}$$

Where:

- $\boldsymbol{x_s}[k]$ is the state vector at time step k
- $\boldsymbol{x_s}[k+1]$ is the state vector at time step $k+1$
- A is the state transition matrix
- G is the process noise gain matrix
- $\boldsymbol{w}$ is the process noise

2) Measurement Model: The measurement model in discrete time can be written as follows:

$$y(k) = H\boldsymbol{x}(k) + v \tag{30}$$

Where:

- $y(k)$ is the measurement vector at time step k
- H is the measurement matrix
- v is the measurement noise

For the *deck detection* and *reflectors detection* methods: $H = [1\ 0\ 0]^T$; for the *Doppler-based*: $H = [0\ 1\ 0]^T$

3) KF Implementation: Using the preceding process and measurement models, the KF algorithm was implemented using the built-in KF class in MATLAB [21].

III. Discussion

The vessels were subject to motions around 2-meter amplitude for heave, 2.5 degrees for pitch, and 1 degree for roll, each with a period of approximately 10 seconds and out of phase. The resulting ground truth values are depicted in Figure 10.

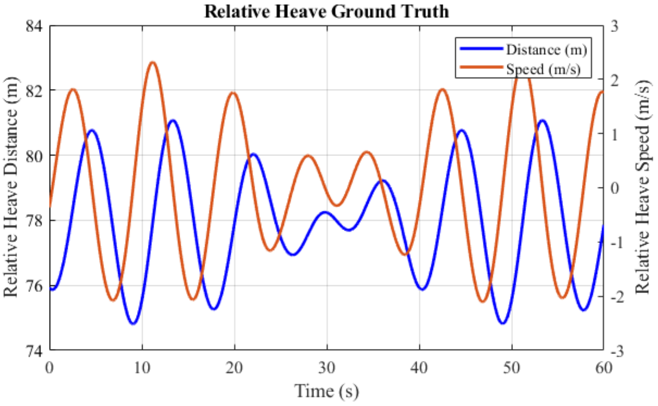


Fig. 10: Illustration of D and $\dot{D}$ reference employed for the verification and evaluation of the implemented solutions.

The errors for each method were calculated by subtracting the estimated relative heave values from the actual values over the simulation period. Additionally, each algorithm computed errors based on five separate runs to ensure accurate and representative results. The resulting error distributions are shown in Figure 11 and Figure 12.

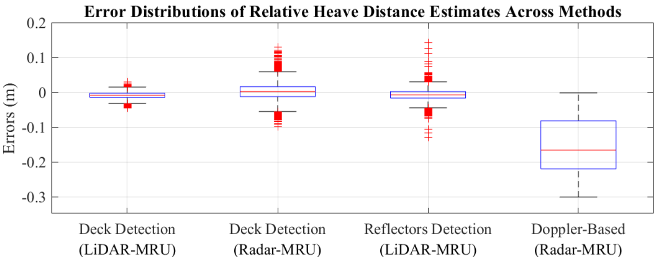


Fig. 11: Error distributions of relative heave distance estimates across the implemented solutions.

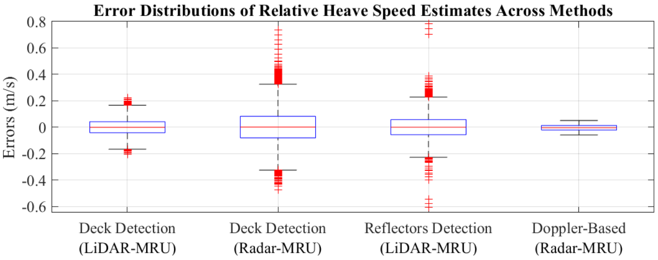


Fig. 12: Error distributions of relative heave speed estimates across the implemented solutions.

An overview of the results achieved with each implemented measurement solution is presented in Table III. A thorough analysis of these results reveals the absence of a singular optimal solution, but rather, the existence of multiple viable alternatives for implementation.

TABLE III: Implemented measuring solutions performance overview

KPI	MAE_D		$MAE_{\dot{D}}$		APT
	$[m]$	$[\%]$	$[\frac{m}{s}]$	$[\%]$	$[ms]$
1. Deck Detection (LiDAR-MRU)	0.010	0.3	0.048	2.1	91.7
2. Deck Detection (RADAR-MRU)	0.018	0.6	0.098	4.3	34.8
3. Reflectors Detection (LiDAR-MRU)	0.013	0.4	0.068	3.0	1.9
4. Doppler-Based (RADAR-MRU)	0.159	5.2	0.018	0.8	2.4

Note: The percentages, $[\%]$, indicate the proportion relative to the maximum amplitude of the relative heave ground truth (Figure 10).

The *reflectors detection* method offers an optimal trade-off between accuracy and processing speed. Its reflectivity-based point filtering maximises processing speed, and it ranks as the second most accurate method in terms of Mean Absolute Error (MAE). However, it is worth noting that the method's effectiveness relies on the presence of reflectors on the feeder vessel, which can be a constraint in some practical cases. The placement and type of retro-reflectors could be a potential research avenue on its own with a practical test case.

Conversely, the *deck detection* method proves to be the ideal choice for a reflector-independent measurement solution. It offers both robust and accurate relative heave measurements for feeder vessels with a discernible ship deck. However, a careful trade-off between accuracy and processing speed is required. Even though the algorithm parameters have been selected carefully, further investigation could focus on improving this trade-off. While the *deck detection* method performed adequately with the RADAR-MRU sensor unit, choosing the LiDAR-MRU sensor unit is advisable. The latter provides enhanced coverage of the supplier vessel, a crucial factor for achieving higher accuracy.

Lastly, the standalone use of the *Doppler-based* method is discouraged, as it cannot measure the absolute relative heave distance, resulting in drift issues, even over brief periods. Nonetheless, it offers a significant advantage by delivering the most accurate relative heave speed measurements. Therefore, it is highly advisable to investigate fusing the *Doppler-based* solution with the others or even combining the *deck detection* and Reflector Detection methods to improve overall accuracy and robustness.

IV. Conclusion

The implemented solutions accurately estimate relative heave distance with an MAE ranging from 0.3% to 5.2% of the reference's maximum amplitude and speed estimates within the range of 0.8% to 4.3%. Notably, these solutions achieved relatively low computational times ranging from 1.9-91.7 $[ms]$, displaying potential to be integrated into real-time applications. The *reflectors detection* (LiDAR-MRU) method achieved the optimal balance between accuracy and processing speed, albeit reliant on reflective markers. The *deck detection* (LiDAR-MRU) method yielded the highest accuracy in distance measurements but came with the slowest processing speed. On the other hand, the *Doppler-based* (Radar-MRU) method excelled in speed measurements but displayed susceptibility to distance estimates drift.

For future research, validating the simulation results using actual sensor data obtained in an up-to-scale setup is advised. Furthermore, fusing the developed methods is highly suggested to achieve more robust and accurate estimates.

Acknowledgments

The authors would like to thank the university supervisors, Vasso and Yusong, for their guidance during this research. Special appreciation goes to the company supervisors, Robert, Jeroen, and Jesse, for their constant support and feedback. The author also expresses gratitude to family and friends for their unwavering encouragement.

References

[1] U. E. I. Administration. (2021) International energy outlook 2021. [Online]. Available: https://www.eia.gov/outlooks/ieo/pdf/IEO2021_Narrative.pdf

[2] H. Díaz and C. Guedes Soares, "Review of the current status, technology and future trends of offshore wind farms," *Ocean Engineering*, vol. 209, p. 107381, 2020.

[3] D. Rippel, N. Jathe, M. Becker, M. Lütjen, H. Szczerbicka, and M. Freitag, "A review on the planning problem for the installation of offshore wind farms," *IFAC-PapersOnLine*, vol. 52, no. 13, pp. 1337–1342, 2019.

[4] U. S. G. A. Office. (2020) Offshore wind energy. [Online]. Available: https://www.gao.gov/assets/gao-21-153-highlights.pdf

[5] L. M. S. Board, *Annual Report 2007-08*. Lake Manitoba Stewardship Board, 2008.

[6] Huisman. (2022) Unpublished internal company document. [Online]. Available: https://www.huismanequipment.com/en/

[7] NOV. (2017) Sealift compensation system. [Online]. Available: https://www.youtube.com/watch?v=oJHl6g9R6z8

[8] MacGregor. (2017) Macgregor ship-to-ship compensation system. [Online]. Available: https://www.youtube.com/watch?v=5_QoSTrU2gc

[9] R. A. McKenzie and R. A. Irani, "Motion compensation for maritime cranes during time-varying operations at the pendulum's natural frequency," *Mechanism and Machine Theory*, vol. 168, 2022.

[10] S. S. Tørdal, "Real-time motion compensation in ship-to-ship load handling," Ph.D. dissertation, Universitetet i Agder, 2019. [Online]. Available: https://uia.brage.unit.no/uia-xmlui/bitstream/handle/11250/2590151/thesis_no_articles.pdf?sequence=1

[11] S. S. Tordal, P.-O. Løvsland, and G. Hovland, "Testing of wireless sensor performance in vessel-to-vessel motion compensation," in *IECON 2016 - 42nd Annual Conference of the IEEE Industrial Electronics Society*, 2016, pp. 654–659.

[12] S. S. Tørdal and G. Hovland, "Relative vessel motion tracking using sensor fusion, aruco markers, and mru sensors," *Modeling, Identification and Control*, vol. 38, no. 2, pp. 79–93, 2017.

[13] S. S. Tordal, W. Pawlus, and G. Hovland, "Real-time 6-dof vessel-to-vessel motion compensation using laser tracker," *OCEANS 2017 - Aberdeen*, vol. 2017-October, p. 1 – 9, 2017.

[14] S. S. Tørdal and G. Hovland, "Ship-to-Ship State Observer Using Sensor Fusion and the Extended Kalman Filter," *Journal of Offshore Mechanics and Arctic Engineering*, vol. 141, no. 4, 2019.

[15] H. K. Holen, A. M. Sjøberg, and O. Egeland, "Estimation of Ship-Deck Motion using LIDAR,Gyroscopes and Cameras," *Modeling, Identification and Control*, vol. 42, no. 3, pp. 99–112, 2021.

[16] E. Balestrieri, P. Daponte, L. De Vito, and F. Lamonaca, "Sensors and measurements for unmanned systems: An overview," *Sensors*, vol. 21, no. 4, 2021.

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

[18] MathWorks. (2023) Auv deep dive - matlab & simulink. [Online]. Available: https://nl.mathworks.com/videos/series/auv-deep-dive.html

[19] J. M. Martínez-Otzeta, I. Rodríguez-Moreno, I. Mendialdua, and B. Sierra, "Ransac for robotic applications: A survey," *Sensors*, vol. 23, no. 1, 2023.

[20] S. Y. Chen, "Kalman filter for robot vision: A survey," *IEEE Transactions on Industrial Electronics*, vol. 59, no. 11, pp. 4409–4420, 2012.

[21] MathWorks. (2017) Linear kalman filter for object tracking. [Online]. Available: https://nl.mathworks.com/help/driving/ref/trackingkf.html

B
Huisman RHC Control Scheme

Figure B.1 provides additional insights relevant to developing a Relative Heave Measurement system. It highlights the significance of incorporating both the measurement of relative heave distance (utilised by the PID Position Controller) and its rate of change (utilised as a feedforward gain) within the Huisman Control Scheme. This integration suggests that a novel measuring solution should provide both inputs to ensure effective load synchronisation with the secondary vessel deck during Ship-to-Ship transfers.

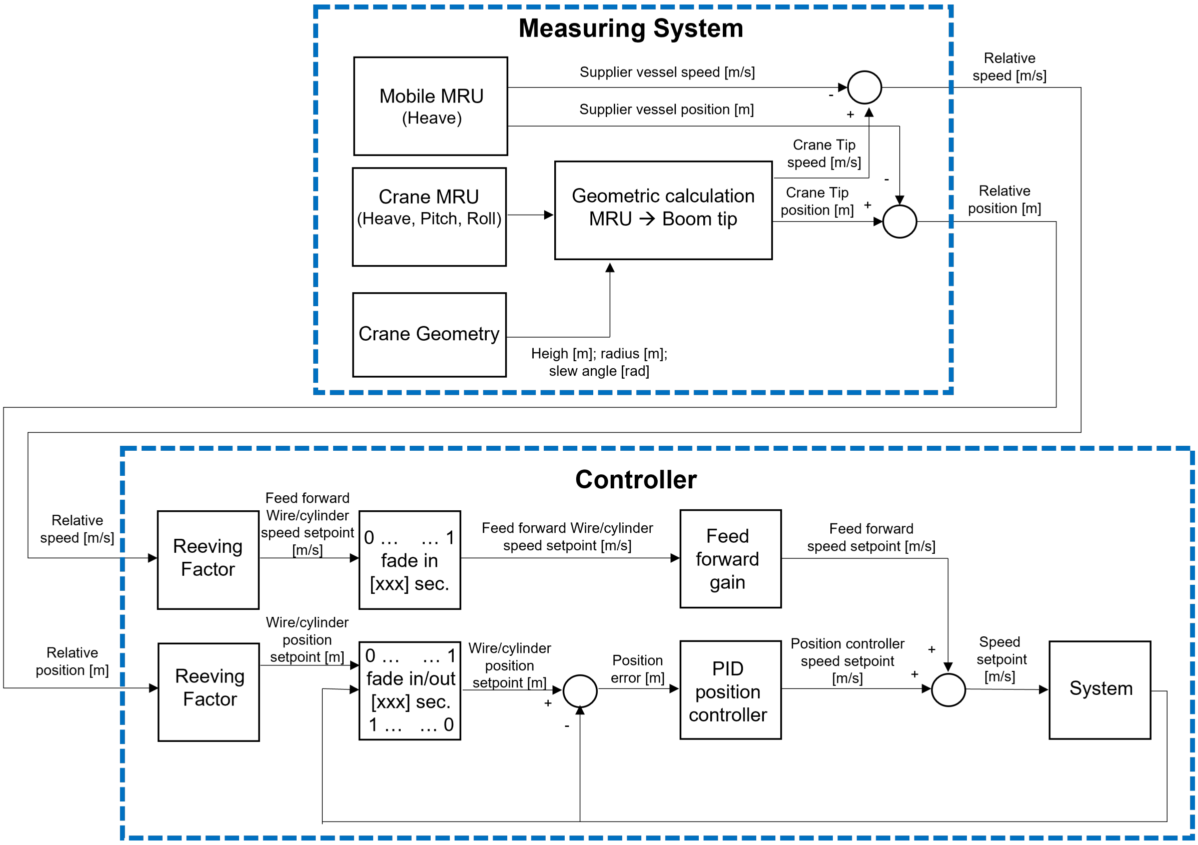

Measuring System
Mobile MRU (Heave)
Supplier vessel speed [m/s]
Supplier vessel position [m]
-
+
Relative speed [m/s]
Crane MRU (Heave, Pitch, Roll)
Geometric calculation MRU → Boom tip
Crane Tip speed [m/s]
Crane Tip position [m]
-
+
Relative position [m]
Crane Geometry
Heigh [m]; radius [m]; slew angle [rad]
Controller
Relative speed [m/s]
Reeving Factor
Feed forward Wire/cylinder speed setpoint [m/s]
0 … … 1 fade in [xxx] sec.
Feed forward Wire/cylinder speed setpoint [m/s]
Feed forward gain
Feed forward speed setpoint [m/s]
Relative position [m]
Reeving Factor
Wire/cylinder position setpoint [m]
0 … … 1 fade in/out [xxx] sec. 1 … … 0
Wire/cylinder position setpoint [m]
+
-
Position error [m]
PID position controller
Position controller speed setpoint [m/s]
+
+
Speed setpoint [m/s]
System

Figure B.1: Huisman's Current Solution for Ship-to-Ship Load Transfer using a Crane MRU placed on the primary ship and a Mobile MRU placed on the Secondary vessel. Figure adapted from [1]

C

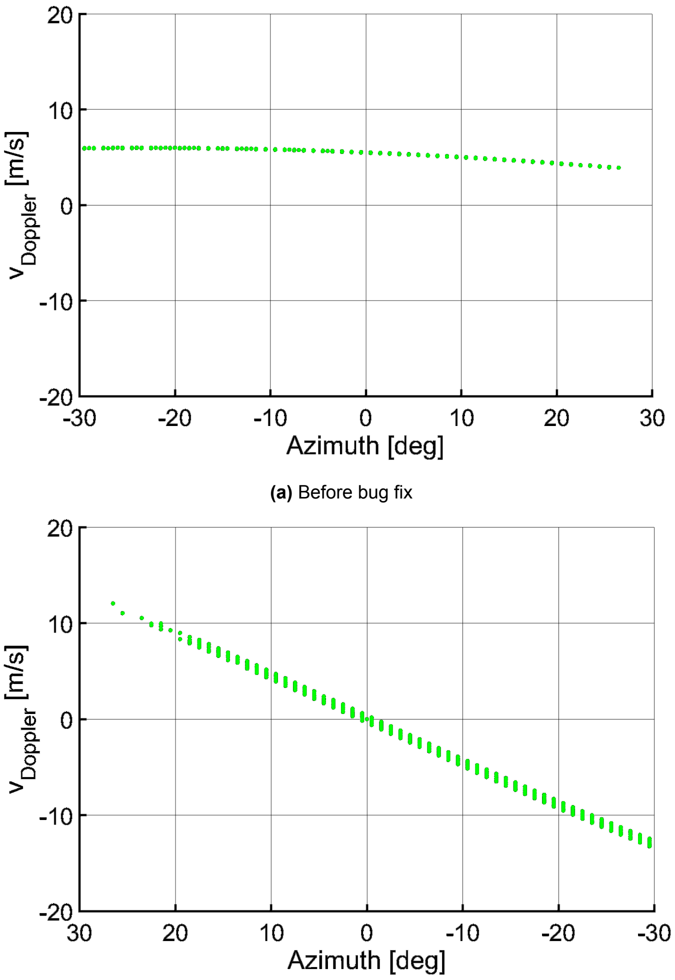
Relative Heave Distance Equation

$$
\begin{aligned}
D = {} & (y_{b_1} c_{\psi_{b2}} s_{\phi_{b2}} - yb2 c_{\psi_{b2}} s_{\phi_{b2}} + x_{b_1} s_{\phi_{b2}} s_{\psi_{b2}} - xb2 s_{\phi_{b2}} s_{\psi_{b2}} + y_{pt} c_{\phi_{b_1}} c_{\psi_{b1}} c_{\psi_{b2}} s_{\phi_{b2}} + z_{pt} c_{\psi_{b1}} c_{\psi_{b2}} s_{\phi_{b_1}} s_{\phi_{b2}} \\
& + x_{pt} c_{\psi_{b1}} c_{\theta_{b_1}} s_{\phi_{b2}} s_{\psi_{b2}} - x_{pt} c_{\psi_{b2}} c_{\theta_{b_1}} s_{\phi_{b2}} s_{\psi_{b1}} + y_{pt} c_{\phi_{b_1}} s_{\phi_{b2}} s_{\psi_{b1}} s_{\psi_{b2}} + z_{pt} s_{\phi_{b_1}} s_{\phi_{b2}} s_{\psi_{b1}} s_{\psi_{b2}} \\
& - z_{pt} c_{\phi_{b_1}} c_{\psi_{b1}} s_{\phi_{b2}} s_{\psi_{b2}} s_{\theta_{b_1}} + z_{pt} c_{\phi_{b_1}} c_{\psi_{b2}} s_{\phi_{b2}} s_{\psi_{b1}} s_{\theta_{b_1}} + y_{pt} c_{\psi_{b1}} s_{\phi_{b_1}} s_{\phi_{b2}} s_{\psi_{b2}} s_{\theta_{b_1}} \\
& - y_{pt} c_{\psi_{b2}} s_{\phi_{b_1}} s_{\phi_{b2}} s_{\psi_{b1}} s_{\theta_{b_1}})/(c_{\phi_{b2}} c_{\theta_{b2}}) - (zb2 c_{\theta_{b2}} - z_{b_1} c_{\theta_{b2}} + x_{b_1} c_{\psi_{b2}} s_{\theta_{b2}} - xb2 c_{\psi_{b2}} s_{\theta_{b2}} \\
& - x_{pt} c_{\theta_{b2}} s_{\theta_{b_1}} - y_{b_1} s_{\psi_{b2}} s_{\theta_{b2}} + yb2 s_{\psi_{b2}} s_{\theta_{b2}} - z_{pt} c_{\phi_{b_1}} c_{\theta_{b_1}} c_{\theta_{b2}} + y_{pt} c_{\theta_{b_1}} c_{\theta_{b2}} s_{\phi_{b1}} \\
& + x_{pt} c_{\psi_{b1}} c_{\psi_{b2}} c_{\theta_{b_1}} s_{\theta_{b2}} - y_{pt} c_{\phi_{b_1}} c_{\psi_{b1}} s_{\psi_{b2}} s_{\theta_{b2}} + y_{pt} c_{\phi_{b_1}} c_{\psi_{b2}} s_{\psi_{b1}} s_{\theta_{b2}} - z_{pt} c_{\psi_{b1}} s_{\phi_{b_1}} s_{\psi_{b2}} s_{\theta_{b2}} \\
& + z_{pt} c_{\psi_{b2}} s_{\phi_{b_1}} s_{\psi_{b1}} s_{\theta_{b2}} + x_{pt} c_{\theta_{b_1}} s_{\psi_{b1}} s_{\psi_{b2}} s_{\theta_{b2}} - z_{pt} c_{\phi_{b_1}} c_{\psi_{b1}} c_{\psi_{b2}} s_{\theta_{b_1}} s_{\theta_{b2}} + y_{pt} c_{\psi_{b1}} c_{\psi_{b2}} s_{\phi_{b_1}} s_{\theta_{b_1}} s_{\theta_{b2}} \\
& - z_{pt} c_{\phi_{b_1}} s_{\psi_{b1}} s_{\psi_{b2}} s_{\theta_{b_1}} s_{\theta_{b2}} + y_{pt} s_{\phi_{b_1}} s_{\psi_{b1}} s_{\psi_{b2}} s_{\theta_{b_1}} s_{\theta_{b2}})/c_{\theta_{b2}};
\end{aligned}
\tag{C.1}
$$

D

MathWorks Bug Fix

The author collaborated with MathWorks and obtained an advanced version of MATLAB specifically designed to address the identified bug. Refer to Figure D.1 for a visual comparison of the Radar output before and after the bug fix. The author's valuable input will be integrated into upcoming releases of MATLAB, positively impacting future research performed with MATLAB's RADAR Toolbox.



(a) Before bug fix

(b) After bug fix

Figure D.1: Comparison of measured velocities before and after the bug fix when only pitch motion is applied. The velocity at the centre of rotation (azimuth=0) should be zero. This condition is satisfied in (b) but not in (a). Additionally, the velocity varies with the arm length, which aligns with expectations.

E

Relative Velocities Equations

Expanding the Equation 4.17.

$$\dot{\boldsymbol{r}}_p^s = \begin{bmatrix} \dot{x}_b \\ \dot{y}_b \\ \dot{z}_b \end{bmatrix} + \begin{bmatrix} -\dot{\psi}c_\theta s_\psi - \dot{\theta}c_\psi s_\theta & \dot{\phi}(s_\phi s_\psi + c_\phi c_\psi s_\theta) - \dot{\psi}(c_\phi c_\psi + s_\phi s_\psi s_\theta) + \dot{\theta}c_\psi c_\theta s_\phi & \dot{\phi}(c_\phi s_\psi - c_\psi s_\phi s_\theta) + \dot{\psi}(c_\psi s_\phi - c_\phi s_\psi s_\theta) + \dot{\theta}c_\phi c_\psi c_\theta \\ \dot{\psi}c_\psi c_\theta - \dot{\theta}s_\psi s_\theta & \dot{\theta}c_\theta s_\phi s_\psi - \dot{\psi}(c_\phi s_\psi - c_\psi s_\phi s_\theta) - \dot{\phi}(c_\psi s_\phi - c_\phi s_\psi s_\theta) & \dot{\psi}(s_\phi s_\psi + c_\phi c_\psi s_\theta) - \dot{\phi}(c_\phi c_\psi + s_\phi s_\psi s_\theta) + \dot{\theta}c_\phi c_\theta s_\psi \\ -\dot{\theta}c_\theta & \dot{\phi}c_\phi c_\theta - \dot{\theta}s_\phi s_\theta & -\dot{\phi}c_\theta s_\phi - \dot{\theta}c_\phi s_\theta \end{bmatrix} \begin{bmatrix} x_{p/b} \\ y_{p/b} \\ z_{p/b} \end{bmatrix} \tag{E.1}$$

Expanding the Equation 4.18.

$$\begin{bmatrix} x_b \\ y_b \\ z_b \end{bmatrix} + \begin{bmatrix} c_{\psi_{b_2}} c_{\theta_{b_2}} & c_{\phi_{b_2}} s_{\psi_{b_2}} + c_{\psi_{b_2}} s_{\phi_{b_2}} s_{\theta_{b_2}} & s_{\phi_{b_2}} s_{\psi_{b_2}} - c_{\phi_{b_2}} c_{\psi_{b_2}} s_{\theta_{b_2}} \\ -c_{\theta_{b_2}} s_{\psi_{b_2}} & c_{\phi_{b_2}} c_{\psi_{b_2}} - s_{\phi_{b_2}} s_{\psi_{b_2}} s_{\theta_{b_2}} & c_{\psi_{b_2}} s_{\phi_{b_2}} + c_{\phi_{b_2}} s_{\psi_{b_2}} s_{\theta_{b_2}} \\ s_{\theta_{b_2}} & -c_{\theta_{b_2}} s_{\phi_{b_2}} & c_{\phi_{b_2}} c_{\theta_{b_2}} \end{bmatrix} \begin{bmatrix} x_{p/b} \\ y_{p/b} \\ z_{p/b} \end{bmatrix} = \begin{bmatrix} cos(\alpha)cos(\beta)r \\ -cos(\alpha)sin(\beta)r \\ sin(\alpha)r \end{bmatrix} \tag{E.2}$$

Solving Equation 4.18 for $\boldsymbol{r}_{p/b}^b = [x_{p/b}, y_{p/b}, z_{p/b}]^T$.

$$\begin{bmatrix} x_{p/b} \\ y_{p/b} \\ z_{p/b} \end{bmatrix} = \begin{bmatrix} z_b s_\theta - x_b c_\psi c_\theta - y_b c_\theta s_\psi + r c_\beta c_\alpha s_\theta + r c_\psi c_\theta s_\alpha - r c_\alpha s_\beta c_\theta s_\psi \\ x_b c_\phi s_\psi - y_b c_\phi c_\psi - z_b c_\theta s_\phi - r c_\phi s_\alpha s_\psi - x_b c_\psi s_\phi s_\theta - y_b s_\phi s_\psi s_\theta - r c_\alpha c_\phi c_\psi s_\beta - r c_\beta c_\alpha c_\theta s_\phi + r c_\psi s_\alpha s_\phi s_\theta - r c_\alpha s_\beta s_\phi s_\psi s_\theta \\ y_b c_\psi s_\phi - z_b c_\phi c_\theta - x_b s_\phi s_\psi - x_b c_\phi c_\psi s_\theta + r s_\alpha s_\phi s_\psi - y_b c_\phi s_\psi s_\theta - r c_\beta c_\alpha c_\phi c_\theta + r c_\alpha c_\psi s_\beta s_\phi + r c_\phi c_\psi s_\alpha s_\theta - r c_\alpha c_\phi s_\beta s_\psi s_\theta \end{bmatrix} \tag{E.3}$$

F

Algorithms

Algorithm 2 RANSAC for Plane Fitting

```
function RANSAC($pointCloud, maxIterations, inlierThreshold$)
    $bestInlierCount \leftarrow 0$
    $bestPlaneModel \leftarrow$ null
    for $iteration \leftarrow 1$ to $maxIterations$ do
        // Randomly select a minimal sample of points
        $samplePoints \leftarrow$ randomlySelectMinimalSample$(pointCloud)$
        // Fit a plane model to the selected points
        $currentPlaneModel \leftarrow$ fitPlaneModel$(samplePoints)$
        $inlierCount \leftarrow 0$
        for each $point$ in $pointCloud$ do
            if $point$ is an inlier to $currentPlaneModel$ then
                Increment $inlierCount$
            end if
        end for
        if $inlierCount > bestInlierCount$ then
            // Update the best model if a better one is found
            $bestInlierCount \leftarrow inlierCount$
            $bestPlaneModel \leftarrow currentPlaneModel$
        end if
    end for
    return $bestPlaneModel$
end function
```

Algorithm 3 Multiple Planes Routine

```
function MultiplePlanes($pointCloud$, $N$)
    $bestMetricResult \leftarrow 0$
    $bestPlane \leftarrow$ null
    $currentPointCloud \leftarrow pointCloud$
    for $iteration \leftarrow 1$ to $N$ do
        // Fit plane to current Point Cloud
        $currentPlane \leftarrow$ pcfitplane($currentPointCloud$)
        // Compute metric result for current plane
        $currentMetricResult \leftarrow$ calcmetric($currentPlane$)
        if $currentMetricResult$ is larger than $bestMetricResult$ then
            $bestMetricResult \leftarrow currentMetricResult$
            $bestPlane \leftarrow currentPlane$
        end if
        $currentPointCloud \leftarrow$ removeInliers($currentPointCloud, currentPlane$)
    end for
    return $bestPlane$
end function
```

Algorithm 4 Reflectors Detection

```
function ExtractReflectivePointCloud($pointCloud$, $intensityThreshold$)
    // Initialize empty point cloud
    $reflectivePointCloud \leftarrow$ null
    for point $\in pointCloud$ do
        // Get intensity of the point
        $intensity \leftarrow$ getIntensity(point)
        if $intensity > intensityThreshold$ then
            // Add point to reflective point cloud
            addPoint(reflectivePointCloud, point)
        end if
    end for
    return $reflectivePointCloud$
end function
```

G
Detailed Vessel Motions

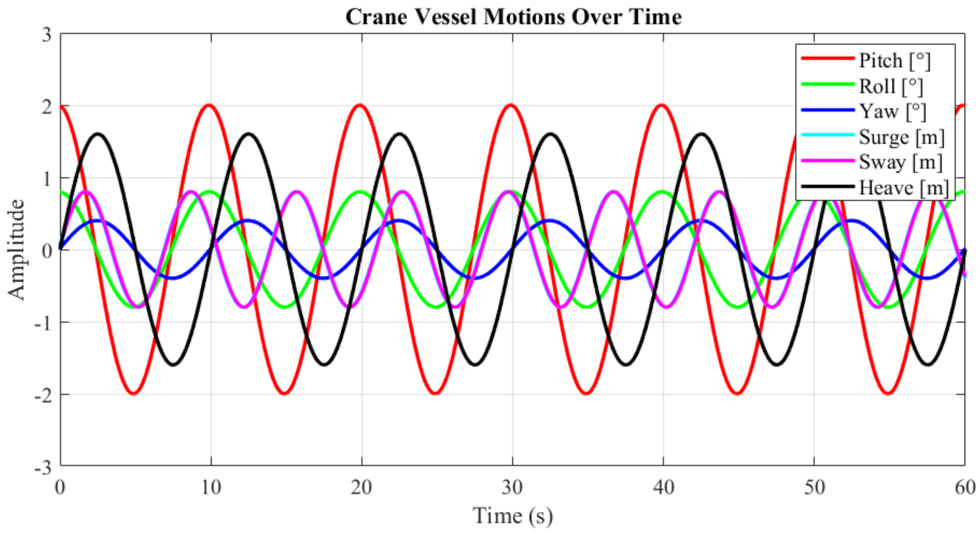


Figure G.1: Crane vessel 6 DOF motions employed in the STS simulation workflow.

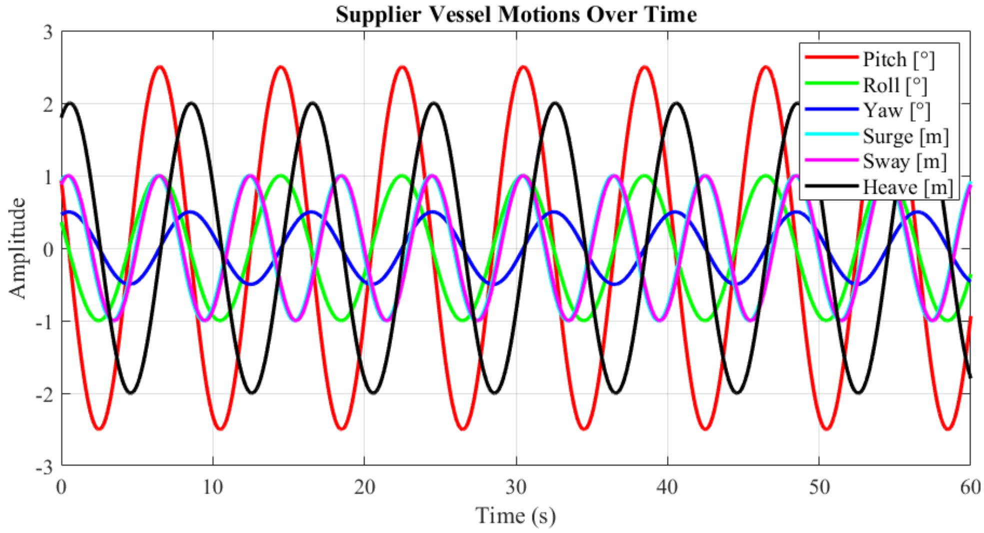


Figure G.2: Supplier vessel 6 DOF motions employed in the STS simulation workflow.