



Title	自動着棧の制御とセンシング技術に関する研究
Author(s)	澤田, 涼平
Citation	大阪大学, 2023, 博士論文
Version Type	VoR
URL	https://doi.org/10.18910/92962
rights	
Note	

The University of Osaka Institutional Knowledge Archive : OUKA

<https://ir.library.osaka-u.ac.jp/>

The University of Osaka

学位論文

自動着棧の制御とセンシング技術に関する研究

澤田涼平

令和 5 年 6 月

大阪大学大学院工学研究科

地球総合工学専攻船舶海洋工学コース

—This page intentionally left for blank—

Research on Control and Sensing Techniques of Automatic Berthing

(自動着桟の制御とセンシング技術に関する研究)

By

Ryohei Sawada

A Dissertation Submitted to
the Department of Naval Architecture and Ocean Engineering,
Graduate School of Engineering, Osaka University,
in Partial Fulfillment of the Requirements for
the Degree of Doctor of Philosophy in Engineering

June 2023

Supervised by
Professor Atsuo Maki

—This page intentionally left for blank—

Abstract

The development of technologies related to automated ships, or MASS (Maritime Autonomous Surface Ship), is currently being actively pursued in many countries around the world. Among the several functions required for MASS, the automation of berthing maneuvers is one of the most important, along with collision avoidance maneuvers. The Maritime Safety Committee (MSC) of the International Maritime Organization (IMO) has conducted an RSE (Regulatory Scoping Exercise) survey of existing regulations. The MSC (IMO: International Maritime Organization) is currently discussing the MASS Code, a set of rules related to operations of MASSs, following the RSE. It is expected that autonomous shipping technologies will support the crew as an extension of the current navigation support systems such as automatic steering systems, even if they do not lead to the ultimate unmanned vessel. The author has recently conducted research on the automation of berthing maneuvers, mainly with the aim of reducing the workload of seafarers, considering the challenges posed by manpower shortage and the aging of seafarers.

This paper deals with the control and sensing technologies for automatic berthing control. The main purpose of the control technology for automatic berthing is to automate approach maneuvers to a pier. Approach maneuvering refers to maneuvering at lower speeds within harbors, distinct from navigation speed, and excludes activities such as rope tethering or mooring until the vessel reaches a parallel stop near a wharf or pier. During a berthing maneuver, the vessel is navigated at low speeds. In certain situations, the propeller may be in idle condition (main engine idling), resulting in relatively poor rudder performance compared to normal navigation speeds. Furthermore, when operating a ship under conditions of reduced rudder power, the impact of disturbance forces on the ship's hull is also relatively significant.

This paper first describes automatic berthing control techniques. An algorithm comprising path planning and path following control is proposed for berthing control. Furthermore, within the harbor, particularly near the pier, the ship motion can be influenced by external disturbances such as ocean currents and winds. In order to reduce the problem of cross track error (CTE) during path following control, a practical 2-DOF control method for berthing controller is proposed in this study. This method can be used in conjunction with the base path-following control to reduce the path error by adjusting the rudder in accordance with the estimated wind loads acting on the hull. This paper also describes the numerical simulations used in the process of developing the control method and presents the results of numerical simulations and shipboard tests to demonstrate the effectiveness of the proposed method.

With the practical application of autonomous navigation technology for not only the automation of berthing maneuvers but also autonomous operation of ships, there is a demand for performance aspects such as safety and stability. In this study, we also discuss the implementation examples of efficient development flow, graphical user interface (GUI) for information provision, introduction of industrial controllers capable of real-time control, and alert systems for safety, along with their implementation examples.

Regarding sensing technology, research results using LiDAR (Light Detection and Ranging or Laser

Imaging Detection and Ranging), which has recently been used in the field of automatic vehicle operation, will be presented to explore its potential application to automatic pier landings. Specifically, we present: 1) path planning for automatic berthing that avoids obstacles around a pier, 2) localization at sea using LiDAR SLAM (Simultaneous Localization and Mapping), and 3) quay wall detection during berthing.

要旨

現在、自動運航船、英語では MASS (Maritime Autonomous Surface Ship) に関する技術開発が世界各国で活発に進められている。自動運航船に求められるいくつかの機能のうち、避航操船に並び重要な技術が着桟操船の自動化機能である。規則面についても、国際海事機関 (IMO : International Maritime Organization) の海上安全委員会 (MSC : Maritime Safety Committee) においても既存規則群の調査である RSE (Regulatory Scoping Exercise) を経て、いわゆる MASS コードと呼ばれる自動運航関連の規則について議論が進められている状況である。操船の自動化の諸技術は、究極的な無人運航船に至るまでなくとも、現状の自動操舵機の延長として船員の乗務を支援することが期待される。著者は、近年船員の人手不足や高齢化といった課題を念頭に、主に船員の負担低減を目的として着桟操船の自動化に関する研究を遂行した。

本論文は、自動着桟に係る制御技術とセンシング技術を扱う。自動着桟の制御技術については、主に桟橋へのアプローチ操船の自動化を目的とする。ここで、アプローチ操船とは計画速力より小さい速力において航行する港内等の操船であって、岸壁や桟橋等の付近へ平行に停船するまでの段階で、ロープ等の綱取り、係船作業を含まない部分を指す。着桟操船は、船舶が低速で航行する状態であり、場合によってはプロペラの遊転する状況（主機アイドリング状態）でのオペレーションも存在するため、通常の速力での航行状態に比べ相対的に舵効きが悪い条件でのオペレーションが要求される。また、そのような舵力が低下した状況での操船では、船体に働く外乱による力の影響も相対的に大きくなる。

本論文では、最初に着桟の自動制御技術について述べる。着桟制御については、経路計画と経路追従制御からなるアルゴリズムを示す。さらに湾内、特に岸壁や桟橋付近においては潮流や風等の外乱が考えられる。外乱の影響により経路追従時の経路偏差 (CTE: Cross track error) が問題となるが、これを低減するために、本研究では主に風外乱に対する着桟操船向けの実用的な 2 自由度制御手法を提案する。本手法は、ベースとなる経路追従制御と合わせて使用することができ、推定する船体の風圧力と釣り合うように当て舵を行うことで、経路誤差を低減するものである。本論文では、制御手法の開発の過程で使用した数値シミュレーションについても述べるとともに、提案手法の有効性について、数値シミュレーションおよび実船試験の結果を示す。

自動着桟操船の自動化に限らず、自動運航船の技術の実用化に伴っては安全面や安定性等の性能も要求されることとなる。本研究では、効率的な開発フロー、情報提供のためのグラフィカルな GUI、リアルタイム制御が可能な産業用コントローラの導入、安全のためのアラートシステムについても、その実装例とともに議論する。

センシング技術については、自動着桟への応用可能性を探るべく、近年車の自動運転の領域で利用が進められている LiDAR (Light Detection and Ranging ないし Laser Imaging Detection and Ranging) を用いた研究成果を示す。具体的には 1. 桟橋周辺における海岸を避けるように動作する自動着桟の経路計画、2. LiDAR SLAM (Simultaneous Localization and Mapping) を用いた海上における自己位置推定、3. 着桟時の岸壁検知について示す。

目次

1 緒論	1
1.1 自動運航船の歴史	1
1.2 研究背景	3
1.3 研究の目的	3
1.4 本研究の貢献	4
1.5 本論文の構成	5
2 先行研究	8
2.1 自動着桟制御	8
2.1.1 自動着桟研究	8
2.1.2 風外乱下における自動着桟の先行研究	9
2.2 システムの実装に関する先行研究	10
2.3 LiDAR SLAM に関する先行研究	10
3 自動着桟制御の対象船とシステム	15
3.1 緒言	15
3.2 対象船の概要	15
3.3 制御アルゴリズムの概要	15
3.3.1 概要	15
3.3.2 経路計画	17
3.3.3 経路追従制御	20
3.3.4 経路追従制御の特性と補正角の導入	25
3.4 実験船の制御システムの構成	27
3.5 センサー	28
3.6 自動着桟制御システム	29
3.6.1 パーソナルコンピュータによる制御	29
3.6.2 PLC による制御	33
3.7 結論	37
4 自動着桟を想定した LiDAR の適用	40
4.1 緒言	40
4.2 岸壁等を考慮した経路計画への適用	41
4.2.1 Hybrid-state A* アルゴリズム	41

4.2.2 LiDAR による計測データを用いた例	42
4.3 海上における点群地図作成と自己位置推定	44
4.3.1 SLAM のアルゴリズム	45
4.3.2 少ない水平な平面特徴量に対する SLAM	47
4.3.3 実験に使用したセンサ, ソフトウェア, 計測機器	49
4.3.4 結果	51
4.4 岸壁検知システムの開発	57
4.5 結言	60
5 自動着桟制御	64
5.1 緒言	64
5.2 数値実験および実船実験の概要	64
5.2.1 試験海域	64
5.2.2 着桟制御の目標設定	65
5.3 数値計算による検証	66
5.3.1 経路追従制御および風外乱補償制御の検証	66
5.4 実船実験による検証	67
5.4.1 経路追従制御による自動着桟制御の検証	67
5.4.2 風外乱補償制御の検証	69
5.5 その他の検証	71
5.5.1 目標方位角の補正に関する検証と考察	71
5.5.2 助走区間を設けた経路計画に関する検証	73
5.5.3 PC と PLC による自動着桟制御の比較	73
5.6 結言	75
6 自動着桟制御に関するその他の検討	77
6.1 安全な自動着桟システム	77
6.2 経路追従制御に関する考察	78
6.2.1 曲線経路形状の考慮	78
6.2.2 斜航状態の考慮	79
6.2.3 経路偏差の直接的な補償	80
6.2.4 シミュレーションによる検討と考察	80
6.3 結言	83
7 結論	85
A 神峰の操縦運動数学モデル	91
A. 1 着桟操船時低速航行状態の操縦運動数学モデル	91
A. 1.1 風圧力と風況のシミュレーション	94
A. 2 流体力微係数とその他のパラメータの同定	96

第 1 章

緒論

1.1 自動運航船の歴史

船舶の輸送の自動化に関する歴史は、1960 年頃から始まる。1960 年代の自動化船の動向については平井 [1] が詳しい。世界初の自動化船と言われているのは、貨物船「金華山丸」(1961 年竣工)である。金華山丸には船橋からの主機の遠隔操作するブリッジコントロール方式と機関部の集中監視制御方式が採用された。1969 年には、機関区域無人化船(M0 船)の第 1 号が就航した [2]。同時期に、コンピュータ技術の導入にともない航法システムや甲板・荷役システム、推進プラントシステムの集中制御も進み、超合理化船が建造されていった。1980 年代には、省力化から省人化へと関心が移り、作業負荷の高い出入港や荷役作業のための自動化が検討された。このように船舶の自動化については、機器の高度化だけでなく船員制度についても検討が必要となる。国内ではその後、高信頼度知能化船のプロジェクトが進められた。この時期には、すでに人工知能(AI:Artificial Intelligence)に関する言及がある [3]。実際、避航操船の自動化に関する研究においては Fuzzy 制御 [4, 5] や、自動着棧においてはエキスパートシステムやニューラルネットワークを用いた研究 [6–8] が見られる。

航行を支援する装置については、20 世紀頃にオートパイロットが普及した [9–11]。1981 年には、ARPA(自動衝突予防援助装置:Automatic Radar Plotting Aids)が 1 万トン以上の全船舶に搭載が義務付けられた。その後、AIS(船舶自動識別装置: Automatic Identification System)や電子海図(ECDIS)についても条件を満たす船舶への搭載が義務付けられた。このような航海機器の充実は、現在の自動運航に欠かせないものとなっている。

近年では、海難事故や船員需給の逼迫化といった課題と、AIS や ECDIS といった通信、技術の発展に伴って自動運航船が注目されている。海外では、MUNIN プロジェクト(2012-2015 年,[12]), AAWA(2015-2018 年, [13]), MAXCMAS プロジェクト(2015-2018 年, [14, 15]), YARA(2017 年, [16]), NOVIMAR(2017 年, [17])など、数多くの自動運航船のプロジェクトが立ち上がった。国内では、日本財團の MEGURI2040 プロジェクトが始まり、5 つのコンソーシアムがそれぞれ内航コンテナ船やカーフェリー、水陸両用船等を対象とした無人運航船の実証実験を実施した [18]。

国際的な議論の動向も、自動運航船の開発の流れを概観する上で重要である。IMO では、1965 年に開催された MSC8 が、船舶の自動化に関する概念が提案された最初の会議であるとされる [19]。この委員会では、「ship automation」という言葉が、「完全な自動化システム」、「部分的な自動化システム」、「遠隔操船」の両方を含む言葉として述べられている。国際海事機関 IMO の第

Table 1.1 The interim degrees of autonomy of MASS by IMO [20]

Degree one	Ship with automated processes and decision support: Seafarers are on board to operate and control shipboard systems and functions. Some operations may be automated and at times be unsupervised but with seafarers on board ready to take control.
Degree two	Remotely controlled ship with seafarers on board: The ship is controlled and operated from another location. Seafarers are available on board to take control and to operate the shipboard systems and functions.
Degree three	Remotely controlled ship without seafarers on board: The ship is controlled and operated from another location. There are no seafarers on board.
Degree four	Fully autonomous ship: The operating system of the ship is able to make decisions and determine actions by itself.

99回では、海上安全委員会(MSC: Maritime Safety Committee)において、自動運航船を(MASS: Maritime Autonomous Surface Ship)の自動化レベルの暫定案が初めて提案され、MSC100[20]で採択された(Table 1.1)。

IMOでは自動運航船の開発を加速させるべく、自動運航船のトライアルに関する暫定ガイドラインも承認された[21]。IMO以外からも、自動運航船に関するガイドラインが各機関から出ている[22–30]。さらに、次のような規則についての議論を実施し、MASSに求められる規則に関する論点整理 RSE(Regularoty Scoping Exercise)をMSC103において完了したと報告している[31]:

- SOLAS (International Convention for the Safety of Life at Sea)
- COLREGs (Convention On the InternationaL REGulations for Preventing Collisions at Sea)
- STCW, STCW-F (Standards of Training, Certification and Watchkeeping for Sea)

この他にも、Search and Rescure(SAR)などの幅広い事項についても、自動化や無人化に伴う対応が検討されている。RSEの完了に伴い、MSCの所掌を超えた議論を行うためのMSC/LEG/FALのJWG(Joint Working Group)を設置し、MASS Codeの策定な議論を進めている段階にある。MSC105時点で、MASS Code策定のロードマップとして、2024年のMSC109における非強制ガイドラインと、続く2025年の強制ガイドラインの採択が予定されている。現在IMOで議論されているMASS Codeは、リスク解析に基づく目標指向型基準(GBS: Goal Based Standards)の体系に沿って構築される。GBSは、「目標(Goal, tier I)」、「機能要件(Functional requirements, tier II)」、「適合検証(Verification of conformity, tier III)」、「船級規則等(Rules and regulations for ships, tier IV)」、「業界標準等(Industrial standards, tier V)」の5つで構成される。GBSを構成する規則群が成立するまでには、まだ時間を要すると考えられる。本論文の現時点では、機能要件が定まっていないため自動運航船のシステム開発者は、認証当局に求められる機能要件や安全性の要求仕様について、リスク分析等を実施しながら開発していくことが求められる。この動きに先駆ける形で、自動運航船の実用化に向けて懸念される安全性の検証のために、国内では様々な組織、機関から自動運航船に関するガイドラインが発行されている[22, 32]。このようなガイドラインは、開発

の上で必要な要求仕様を決める一助になると考えられるが、依然として開発者側が検討するべき項目は多い。

1.2 研究背景

着桟操船は船員に対して負荷の高い作業のひとつである[33]。この要因として、係船を含めた着桟の作業において必要な人数が多いこと以外に、船の運動特性に起因するものがある。着桟時の船舶は、航海速力に比べて著しく低速の状態で航行することとなる。船舶の低速時操縦運動においては、航海速力で航行している状況に比べて舵が受ける有効流入速度が減じるため、舵効きが悪化する。また、低速時の運動においては、相対的に船体に働く外乱の影響が大きくなる。着桟操船においては、港内の浅水域の近傍や岸壁等の障害物の近くを航行することになるため、座礁や衝突といった事故の可能も高くなる。同じ自動運航船のタスクである避航操船の自動化と比較して、大きく異なる特性を有していると言える。さらに、着桟操船においては、速力の変化が大きく、プロペラや舵の力学についてもプロペラ正転時のみならずプロペラ逆転時の取り扱いについても注力する必要がある。プロペラ逆転時には、正転時と比較し非常に複雑な船尾流場の影響があるため、運動の予測が難しくなる。これとともに、自動制御の観点からは、主機等のアクチュエータの過度な負荷を避けるように設計する必要があり、数学的な制御則の設計理論に加えて、安全性の観点も重要である。本研究は、このような負荷の高い複雑な操船を要求する着桟作業を自動化により支援する目的で、着桟制御技術およびLiDAR(Light Detection and Ranging)またはLaser Imaging Detection and Ranging)を用いた自動着桟のためのセンサ技術に関する研究を実施した。

1.3 研究の目的

本研究の目的は、自動化技術の導入による船員の負荷低減である。そのために、船員にとって最も負荷の高い作業の一つである着桟操船を研究の対象とする。ただし、本研究では、自動化技術を導入することによる船員の負荷低減効果の定量的な評価については実施しない。あくまで、船員の負荷が高い操船作業を自動化するための要件を満たす自動化システムならびに自動化に資する技術について、開発と性能の評価を実施する。人間系に対する評価は実施しないものの、ユーザビリティを考慮したシステムの設計については広く検討した。たとえば、制御状況や機器の状態を監視するためのグラフィカルユーザーインターフェース(GUI, もしくは単にUI), 異常発生時に乗組員へ操船を引き継ぐための警報システムなどについて検討を行う。また、基本的な着桟操船についても、単純な制御の性能のみにとどまらず、より安全な桟橋へのアプローチ角度についても検討をおこなう。提案する自動化システムは、原則的に船上に乗組員が存在するような自動化レベルの船舶における操船支援機能を提供するものとして開発される。将来的には、本提案システムを拡張することで、より高度な自動化レベルへの適用の足がかりとなるように設計しているが、現状、船舶の機器の安全対策や堅牢性は、人の保守や監視なしに成立しないため。このような自動化レベルを対象とした。したがって、運航設計領域(ODD: Operational Design Domain)に代表される、自動化システムが対象とする航行条件を逸脱する可能性を考慮し、適切にシステムが人へ操船権を移譲できるような設計としている。

一方で、現状の船舶に搭載される機器の性能限界やセンサ系の冗長化についても、本研究では一側面からのアプローチを実施する。それは、近年自動車の自動運転技術でも利用される LiDAR センサを用いた研究である。LiDAR は、レーザ光を用いてセンサの周囲環境を計測するセンサである。LiDAR の詳細はのちの章で説明される。LiDAR は、製品によりその測距距離は大きく異なるが、周囲 360 度を 3 次元的に計測する一般的な 3D LiDAR の製品は 100~200 [m] 前後の範囲を計測することができる [34]。中には 500 [m] 以上測距可能なセンサも存在するが、距離が大きくなると計測できる点群密度が小さくなる。点群密度が小さくなると、遠くにある小さな反射面積を持つ物体を検知することが難しくなるため、多くの製品は、点群密度を維持するために計測範囲を絞ることになる。このような特性から、自動車とは異なり、数マイル先を見て航行する船舶への適応は難しいとされてきた。一方で、港内操船や着桟操船においては、港外と比較して周辺の障害物や船舶が比較的自船の近くに存在するため、一般的な 3D LiDAR でも十分に活用することができる。3D LiDAR の船舶の応用についてはいくつかの場面が考えられるが、本研究では障害物、特に岸壁の計測点群を活用した、経路計画、姿勢計測（自己位置推定、方位角の推定を含む）、岸壁の検知と距離の計測について研究を行った。

1.4 本研究の貢献

本研究の貢献は次の通りである。まず、自動着桟技術に関して、基本的な構成として経路計画と経路追従制御からなる自動着桟アルゴリズムを提案する。本アルゴリズムは、主要な 2 つの機能により分割することができるため、着桟経路の生成と制御問題を分けて設計することができる利点を有する。経路計画については、パラメトリックな曲線である Bézier 曲線を用いた高速で軽量な手法を検討した。経路追従制御については、より広い船舶への適応を念頭に最小限のアクチュエータを要求するという条件を満たす制御手法を検討した。ここでは、pure pursuit と PD 制御から構成される自動操舵則を組み合わせた手法である PPA (Pure Pursuit and Autopilot) 制御を提案する。本論では、経路計画および経路追従制御に関する本提案手法を、数値シミュレーションならびに実船実験において詳細に検証した。数値シミュレーションにおいては、使用する船舶の低速時の大斜航角を有する運動を表現できる操縦運動数学モデルを構築し、これを利用して提案手法の検証を実施した。この操縦運動数学モデルの詳細については、末尾の付録にモデルならびにパラメータの同定手順を記載している。

さらに、PPA 制御に加えて、風外乱に対応した制御器である FFPPA (Feed-Forward PPA) 制御を提案する。本制御は、船舶が運動中に船体に受ける横方向の風圧力を推定し、これに相当する舵力を計算することで、風圧力による運動の行き足が発達する前にその影響に対応するものである。本制御則を導くために船体運動を 3 自由度の状態方程式で定式化し、得られた線形化した横方向舵力モデルに基づいて、フィードフォワード制御を構成する。最終的にフィードバック制御で構成される PPA 制御に対して、風外乱に対応するフィードフォワード制御則を加えた 2 自由度制御として FFPPA 制御を導入する。本手法に関しても、数値シミュレーションならびに実船実験において詳細に検証される。

また、制御アルゴリズムを含めた自動運航に必要な機能を一つのシステムとして実装する方法についても示す。多くの自動運航船研究においては、アルゴリズムの評価に限られたものや、システムの概念図を示すにとどまったものが多く、本研究では、現在の最新の MASS の議論に沿つ

た，安全なシステム構築のための，実装の具体例を示し，その動作を確認する．ここでは，自動着棧中に乗組員が非常時対応することを想定した，監視機能，警報機能，ヒューマンフォールバック，MRM (Minimum Risk Maneuver) が含まれる．

これに関連して，自動運航の安全性を向上させるためには，センサ系の冗長化も重要な課題である．本研究で用いる実験船にGNSS (Global Navigation Satellite System) を搭載し，測位と方位角の計測を行っているが，通信状況や海域によって，満足な精度がえられ

1.5 本論文の構成

本論文は第1章から第7章および付録により構成される．次の第2章では，本研究に関連する先行研究についてまとめる．対象として，自動着棧制御技術，システムの実装，LiDAR SLAMの研究を取り上げる．第3章においては，自動着棧制御の概要と制御システム及びアルゴリズムについて述べる．特に，自動着棧システムのハードウェアおよびソフトウェアの構成，そして風外乱下におけるリアルタイム動作可能な制御則の構成について示す．第4章では，自動着棧を想定したLiDARの適用について述べる．LiDARの着棧操船への応用例として，経路計画，自己位置推定，岸壁検知の研究結果を示す．第5章では，自動着棧の数値シミュレーションおよび実船実験の結果を示す．第6章では，自動着棧制御に関するその他の検討項目について述べる．第7章では，本論文を総括する．付録に，シミュレーションに用いた操縦運動数学モデルについて記載する．

第1章の参考文献

- [1] 平井忠, 自動化船の推移と動向, 日本舶用機関学会誌 **23**(8), 459 (1988)
- [2] 安藤恒利, M0 船の概要 (<特集>省エネルギー), 航海 **66**, 15 (1980)
- [3] 飯島幸人, 林尚吾, 高度知能化船の無人航行の研究 (<特集>iain'88, 航海 **97**, 9 (1988))
- [4] 岩崎寛希, 原潔, あいまい推論を用いた避航操船モデル, 日本航海学会論文集 **75**, 69 (1986)
- [5] 長谷川和彦, 上月明彦, Fuzzy 制御による自動避航システムに関する研究, 関西造船協会誌 **205**, 1 (1987)
- [6] H. Yamato, T. Koyama, T. Nakagawa, Automatic berthing using the expert system, IFAC Proceedings Volumes **25**(3), 173 (1992), iFAC Workshop on Artificial Intelligence Control and Advanced Technology in Marine Automation (CAMS '92), Genova, Italy, April 8-10
- [7] 長谷川和彦, 木寺啓治, ニューラルネットワークと知識ベースを併用した自動着桟システム, 関西造船協会誌 **220**, 135 (1993)
- [8] K. Hasegawa, On harbour manoeuvring and neural control system for berthing with tug operation, Proceedings of International Conference Manoeuvring and Control of Marine Craft (1994)
- [9] T.I. Fossen, Recent developments in ship control systems design, World Superyacht Review (1999)
- [10] 長谷川和彦, 安全運航のための計測制御, 計測と制御 **37**, 762 (1998)
- [11] 松田, 船舶用オートパイロット, 計測と制御 **50**(6), 367 (2011)
- [12] MUNIN (Maritime Unmanned Navigation through Intelligence in Networks). URL <https://cordis.europa.eu/project/id/314286/reporting>. Accessed June 2023
- [13] AAWA. URL <https://www.rolls-royce.com/media/press-releases/2016/pr-12-04-2016-aawa-project-introduces-projects-first-commercial-operators.aspx>. Accessed June 2023
- [14] Project MAXCMAS (MAchine eXecutable Collision regulations for Marine Autonomous Systems). URL <https://www.solent.ac.uk/research-innovation-enterprise/research-at-solent/projects-and-awards/project-maxcmas>. Accessed June 2023
- [15] J. Varas, S. Hirdaris, R. Smith, P. Scialla, W. Caharija, Z. Bhuiyan, T. Mills, W. Naeem, L. Hu, I. Renton, D. Motson, E. Rajabally, Maxcmas project: Autonomous COLREGs compliant ship navigation. in *Proceedings of the 16th Conference on Computer Applications and Information Technology in the Maritime Industries (COMPIT) 2017* (2017), pp. 454–464
- [16] Yara birkeland. URL <https://www.yara.com/knowledge-grows/game-changer-for-the-environment/>. Accessed June 2023
- [17] NOVIMAR. URL <https://novimar.eu/>. Accessed June 2023
- [18] 無人運航船プロジェクト「MEGURI2040」. URL <https://www.nippon-foundation.or.jp/what/projects/meguri2040>. Accessed June 2023
- [19] M. Jo, A. Lee, Y. Kim, J. Seo, Study on the potential gaps and themes identified by IMO Regulatory Scoping Exercise(RSE) for the use of Maritime Autonomous Surface Ships(MASS), IOP Conference Series: Materials Science and Engineering **929**(1), 012014 (2020)
- [20] International Maritime Organization. MSC100/20/add.1 Annex 2 framework for the regulatory

- scoping exercise for the use of maritime autonomous surface ships (MASS) (2018)
- [21] International Maritime Organization. MSC.1/Circ.1604, Interim Guidelines for Mass Trials (2019)
- [22] Class NK. 自動運航、自律運航に関するガイドライン (ver.1.0) (2020)
- [23] Class NK. 自動運航船の社会実装へ向けて (2023)
- [24] American Bureau of Shipping (ABS). Advisory on autonomous functionality (2020)
- [25] American Bureau of Shipping (ABS). Requirements for autonomous and remote control functions (2022)
- [26] Korean Resister. Guidance for autonomous ships (2022)
- [27] Lloyd's Register (LR). Unmanned marine systems code (2017)
- [28] DNV GL. Class guideline DNVGL-CG-0264, autonomous and remotely operated ships (2018)
- [29] Norwegian Maritime Authority. Guidance in connection with the construction or installation of automated functionality aimed at performing unmanned or partially unmanned operations (2020)
- [30] VTMIS. The EU operational guidelines on the safe and secure testing environments for maritime autonomous surface ships (MASS) (2020)
- [31] International Maritime Organization. MSC.1-Circ.1638, Outcome of the Regulatory Scoping Exercise for the Use of Maritime Autonomous Surface Ships (MASS) (2021)
- [32] 国土交通省. 自動運航船に関する安全ガイドライン (2022)
- [33] 角田領, 安藤英幸, 大和裕幸, 宮崎恵子, 宮脇恵治, 離着桟操船における船員の作業負担分析, 日本船舶海洋工学会論文集 **6**, 289 (2007)
- [34] A. Carballo, J. Lambert, A. Monrroy, D. Wong, P. Narksri, Y. Kitsukawa, E. Takeuchi, S. Kato, K. Takeda, LIBRE: The multiple 3D lidar dataset. in *2020 IEEE Intelligent Vehicles Symposium (IV)* (2020), pp. 1094–1101

第 2 章

先行研究

2.1 自動着桟制御

2.1.1 自動着桟研究

ここでは、自動着桟に関する先行研究を列挙する。ただし、風外乱への対応を含むより高度な手法については次節で改めて列挙する。

着桟操船の自動化に関する多くの研究において、比例・微分（PD）制御器、ニューラルネットワーク、最適制御理論、進化戦略（ES: Evolution Strategy）などの異なる手法を用いた様々なアプローチが提案されている。Ahmed らは、フィードフォワードニューラルネットワークと PD 制御の heading controller を組み合わせたコントローラを提案した [1]。この研究では、最小限の時間で着桟制御を人工ニューラルネットワーク（ANN: Artificial Neural Network）の学習するために、非線形計画法（NLP: Nonlinear Programming）を用いて作成した学習データを作成している。本研究では、全長 300m を超える VLCC タンカーの実船模型を用い、着桟を目指しており、コントローラに要求される最終着桟位置の精度は $\pm 1.5L$ （実船スケールで 450m 以上）であった。一方、論文では計算時間について触れていないが、模型船を用いた実験ではリアルタイムで制御できていることから、計算時間については高速であると考えられる。この研究以外にも、水野らはスライディングモードという制御方法を用いた自動着桟システムを提案している [2]。着桟時の幅寄せ操船の経路については、SCGR (Sequential conjugate-gradient-restoration) 法により得られた最短時間操船の解をエルミート補間したものを利用している。本手法は可変ピッチプロペラを搭載した練習船を用いた実船実験により検証された。庄司らは、2 点の非線形境界値問題として自動着桟を定式化し、非線形最適化を用いて最短時間のアプローチ問題として舵、主機とサイドスラスターを使った制御の解を得ている [3]。岡崎らは、同様に自動着桟制御を最短時間制御問題として扱い、操船パターンの分類を試みている [4]。また、岡崎らは最適化手法を使わない方法として、ニューラルネットワークを使った研究も行っている [5]。奥山らは、内航船のための最短時間港内操船システムを提案し、その手法を宇部港の実船を用いた着桟操船に適用した [6]。奥山らは、1995 年に大津ら [7] が解いた SCGRA (sequential conjugate gradient-restoration algorithm) を用いて、最短時間操船の解を得ている。提案手法を着桟前の港湾操作に適用し、最適化により得られた解に基づいて船長により操船を行うことで検証した。牧らは、進化型アルゴリズムの一つである共分散行列適応進化戦略（CMA-ES: Covariance Matrix Adaptation Evolution Strategy）を用いて、1 軸 1 舵船の自動着桟問題に対する最適解を獲得している。シミュレーションで得られた最適解

は、最終状態ベクトルと終了条件ベクトルの L1 ノルムが 10^{-4} のオーダーの精度であった。しかし、最適解を求めるのに時間がかかるため実時間での制御には直接適用はできないが、CMA-ES の解はオンライン制御のための最適化問題の初期推測のためのオフライン解として利用することができるとしている。また、船舶の操縦運動の非線形性が着桟問題を複雑にしている原因の一つであると指摘している。したがって、シミュレーション時の運動モデルの非線形性の再現については特別に注意を払う必要があり、これが TCS (Track Control System) の型式認証におけるシミュレーション用の運動モデル [8] や避航操船の自動化でよく用いられる KT モデルといった線形の運動モデル (e.g. [9]) を着桟運動の再現で使うべきではない理由である。

2.1.2 風外乱下における自動着桟の先行研究

風外乱下における着桟の自動化に関する研究も数多く存在する。Ahmed et al. [10] は、ANN と PD 制御を組み合わせたコントローラを提案し、ある風況下にある数値シミュレーションの環境においてタンカーの ESSO Osaka の模型スケールの計算において性能を検証した。実験の結果、ニューラルネットワークによる制御だけでは風の影響に対して制度の高い制御ができず、PD 制御が不可欠であることが報告された。Shuai らは、静的および変動風速の条件下で ANN (Artificial Neural Network) ベースのコントローラを使用した数値シミュレーションの結果を報告している [11]。彼らの研究では、風向についてはドックから離れる方向の条件でのみ検証をしている。静的な風況条件下では、船は安定した航跡でドックにアプローチ操船をしているが、変動風の状況においては船は風下側に流されている。水野と窪島は、3 層の再帰ニューラルネットワーク (RNN: Recurrent Neural Network) を自動着桟制御に適用している [12]。彼らは、可変ピッチプロペラを搭載した 425GT の実船を用いて実験を実施した。水野らは自動着桟制御のための準リアルタイム最適化手法を提案している [13]。この手法は、multiple shooting アルゴリズムとモデル予測制御 (MPC: Model predictive control) をベースとした非線形な着桟軌道の最適化により構成されている。この手法も実船実験によりその性能が検証されている。本手法を用いた場合、着桟までの最適経路を計算するのに必要な時間は 3 分から 5 分となっており、制御開始前に風外乱で船が流されている。また、Martinsen らは、Autonomous Surface Vehicles (ASVs) の自動着桟制御について報告している [14]。彼らは、非線形最適制御問題を定式化し、この問題を軌道追従型の動的位置決めコントローラに適用した。1~3 [m/s] の比較的穏やかな風況で、小型 ASV を用いて手法を検証した。非線形最適化手法を自動着桟に適用した他の成果として、牧ら [15] や秋元ら [16] の研究が挙げられる。これらの研究では、CMA-ES 法が使われている。先行研究で使用された実験船のアクチュエータに注目すると、CPP プロペラを用いた例が多い。CPP は連続的に翼角を変更することができるため、固定ピッチプロペラの船と比較して速度制御が容易である。固定プロペラを持つ実船を使った研究は、著者らの研究を除いてほとんどない [17]。Han らは、無人探査機 (USV: Unmanned Surface Vehicle) の自動着桟制御のために、Extended Dynamic Window Approach (EDWA) を用いた運動計画手法を提案した [18]。この手法は、USV の運動、障害物といった制約や風圧力の影響を考慮し、最適軌道を与える。Han らは風速と風向を変えた数値シミュレーションによるケーススタディを実施した。Piao らは、造風機を用いて風外乱下の模型実験を行った [19]。Piao らは、風外乱下の制御を行うために、Line of Sight (LOS) アルゴリズムを用いた経路計画と Active Disturbance Rejection Control (ADRC) を適用した。この手法を用いて、1.5[m/s] の風外乱条件下で実験を行ったところ、従来の PID 制御器は目標の着桟エリアに到達することがで

きなかったが、ADRC は、模型船を目標の着桟エリアまで制御することができた。清水らは、教師あり学習 (SL: Supervised Learning) と強化学習 (RL: Reinforcement Learning) を用いて、着桟の自動制御を行った [20]。この方法は、着桟への衝突の安全リスクを考慮したもので、風外乱を考慮した模型船スケールの数値シミュレーションにより検証された。Liu らは、自動着桟制御のためにバリアリアノフ関数 (BLF) を用いた方法を提案した [21]。

風外乱下の着桟時の船体運動を評価するためには、シミュレーション空間における風況、特に風速や風向が変動する変動風の計算方法が重要である。最も単純な変動風のシミュレーション手法として、風況の観測値の時系列に基づき、解析により得られた統計量に基づいた確率分布により、乱数を用いて生成する方法がある。風速については Monahan によればガウス分布 [22]、風向に関しては Weibull 分布 [23] などが提案されている。このような方法は、簡易かつ高速に観測値の統計量に基づいた風況を再現できる一方で、各時刻の連続する風況が i.i.d. であるという特徴がある。風は、スペクトラムが幅広い周波数領域にその分布を有するため [24]、海洋波の再現で用いられる重ね合わせの原理を用いた方法を取りうる場合がある。現実的な i.i.d. でない風の時系列を再現する方法として、確率過程に基づくものがある。例えば、Nichita は風のスペクトラムを 2 次元のフィルターを用いて近似し、風の時系列を計算する手法を提案している [25]。また、CARMA(Continuous Auto-Regressive Moving Average) モデルによる近似したスペクトルを、2 次元の確率微分方程式 (SDE: Stochastic Differential Equation) で表現する研究がある [26]。さらに、牧らは 1 次元の確率微分方程式で風を近似することで解析解を求め、高速に不規則な風の時系列を生成する手法を提案している [27–29]。

2.2 システムの実装に関する先行研究

本研究では提案手法を使った自動着桟システムを例に、自動運航船のシステムを実装する方法についても提示する。国内では、日本財團の MEGURI2040 プロジェクトにおいて、自動着桟制御を含む自動運航システムの実船による実証実験がいくつか実施された [30, 31]。しかし、これらの研究では着桟機能に対するシステムの状態監視やアラート管理の実装については触れられていない。関連する研究として、Yong らがモバイルハーバーのための自動ドッキングシステムを提案した [32]。このドッキングシステムは、通常時のパッシブモードと、緊急時の船舶と移動港のドッキングを可能にするエマージェンシーモードを備えている。ここで、実際の社会実装の例として、TCS (Track Control Systems) の規格を挙げたい [8]。IEC62065 は、国際海事機関 (IMO) が採用した TCS の性能基準に適合する最小限の運用・性能要件と試験を定義したものである。この中では、装置間のデータフローや、使用のためのドキュメントの提供などが定められている。また本規格の中では、緊急時や故障時に適切な警報を発することがシステムに要求されている。

2.3 LiDAR SLAM に関する先行研究

LiDAR と LiDAR を用いた SLAM (LiDAR SLAM) は、移動ロボットや自動車の自律走行などに広く利用されている。一方、船舶における LiDAR SLAM は、いくつかの困難があるため研究例が少ない。海上での SLAM は、Ødven が行っている。この例では、2D SLAM である Hector SLAM[33] と 2 つの 3D SLAM アルゴリズムである LOAM (LiDAR Odometry and Mapping[34])

と BLAM (Berkeley Localization and Mapping, [35]) が 16 ライン LiDAR である Velodyne VLP-16 を用いて港内での位置決めと地図作成を実施した。この研究では、RTK-GPS (Real Time Kinematic GPS) 測位と GPS コンパスの方位角と比較した場合の位置と方位角の推定誤差を示して。しかし、いずれの手法も航行に必要な、十分な精度の点群地図を作成することができていない。その結果、Ødven は、どちらの SLAM アルゴリズムも、自律型フェリーの運航のための手法として機能するには不十分であると報告している。

別の研究では、Shan らがアムステルダムの運河を 3 時間航行した際の計測値を LIO-SAM の検証用データセットとして使用した [36]。運河は航路の両側が岸壁に囲まれており、LiDAR の測定距離が小さかった（運河の幅は約 20~100 [m] であった）。この研究では、沖合から岸壁を計測することで SLAM を行った。Shen らはその研究の中で、運河における SLAM がいくつかの理由で難しい問題であることを指摘している。実際、他の陸上の計測データセットと比較して上下方向の特徴量の変化が少ないため、最大旋回速度も小さいにもかかわらず、SLAM の成功率は最も低く、比較に用いた 4 つの SLAM アルゴリズムのうち 3 つは意味のある結果を出すことができていない。運河での SLAM が困難な理由の一つは、地面がないため水面上の平面特徴点が著しく少なく、そのため、垂直方向の点群を統合することが困難であることである。実際、後述の海上 SLAM では、パラメータ設定によっては、垂直方向のドリフトが発生し、地図作成が不可能な地点があった。この課題を解決するため、本研究では LiDAR でスキャンした点群を点群地図上の座標に変換するスキャンマッチングの計算において、座標変換の計算に制約を加えることで点群のマッチング誤差を低減させている。

また、Zhou らは、地上と水面を含むマルチシーン環境に対する S4-SLAM アルゴリズムを提案している [37]。Zhou らは S4-SLAM アルゴリズムを用いて、港湾データセットにおいて GPS と慣性航法装置 (INS) を使用しない結果を示している。この研究において報告されているように、LOAM アルゴリズムはここでも一部の港湾データで失敗している。しかし、彼らの研究では、航跡と最終的な位置のドリフト誤差のみが示されており、方位角推定に関する評価はなされていない。またこの論文で報告されている、S4-SLAM のループクロージャーが機能しなかった問題は、どの港湾データセットでも船の航跡がループしていなかったことが原因の一つと考えられる。ループクロージャーとは、SLAM 時に行われる点群マッチングの後処理であり、作成中の地図全体を歪みを除去する処理を指す。その結果、数 km の計測で 0.56~0.93% (論文中の船の総移動距離に対する偏差の割合で表したもの) のドリフトが発生している。

上記の先行研究を踏まえ、本研究では、実験船が停泊している因島マリーナから北部の高速船乗り場付近までのテストエリアで LiDAR と IMU (慣性計測装置) での計測を行い、海上での LiDAR SLAM の可否を検証する。

第2章の参考文献

- [1] Y.A. Ahmed, K. Hasegawa, Consistently trained artificial neural network for automatic ship berthing control, *TransNav, the International Journal on Marine Navigation and Safety of Sea Transportation* **9**(3), 417 (2015)
- [2] N. Mizuno, S. Matsumoto, Design and evaluation of simple ship's automatic maneuvering system using sliding mode controller, *IFAC Proceedings Volumes* **46**(33), 67 (2013), 9th IFAC Conference on Control Applications in Marine Systems
- [3] K. Shouji, K. Ohtsu, S. Mizoguchi, An automatic berthing study by optimal control techniques, *IFAC Proceedings Volumes* **25**(3), 185 (1992)
- [4] T. Okazaki, K. Ohtsu, N. Mizuno, A study of minimum time berthing solutions, *IFAC Proceedings Volumes* **33**(21), 135 (2000), 5th IFAC Conference on Manoeuvring and Control of Marine Craft (MCMC 2000), Aalborg, Denmark, 23-25 August 2000
- [5] 岡崎忠胤, 正司公一, 水野直樹, 大津皓平, ニューラルネットワークによる最短時間操船に関する研究, *日本航海学会論文集* **97**, 155 (1997)
- [6] E. Okuyama, K. Igarashi, H. Oda, K. Miyazaki, K. Ohtsu, T. Okazaki, Guidance control of vessels using minimum time control, *2007 IEEE International Conference on Systems, Man and Cybernetics* pp. 3650–3655 (2007)
- [7] K. Ohtsu, K. Shoji, T. Okazaki, Minimum time maneuvering of ship with wind disturbances, *IFAC Proceedings Volumes* **28**(2), 338 (1995), 3rd IFAC Workshop on Control Applications in Marine Systems, Trondheim, Norway, 10-12 May
- [8] I.E. Commission. Iec 62065:2014, maritime navigation and radio communication equipment and systems - track control systems - operational and performance requirements, methods of testing and required test results (2014). <https://webstore.iec.ch/publication/6431>
- [9] R. Sawada, K. Sato, T. Majima, Automatic ship collision avoidance using deep reinforcement learning with LSTM in continuous action spaces, *Journal of Marine Science Technology* **26**, 509 (2021)
- [10] Y.A. Ahmed, K. Hasegawa, Automatic ship berthing using artificial neural network trained by consistent teaching data using nonlinear programming method, *Engineering Applications of Artificial Intelligence* **26**(10), 2287 (2013)
- [11] Y. Shuai, G. Li, X. Cheng, R. Skulstad, J. Xu, H. Liu, H. Zhang, An efficient neural-network based approach to automatic ship docking, *Ocean Engineering* **191**, 106514 (2019)
- [12] N. Mizuno, R. Kuboshima, Implementation and evaluation of non-linear optimal feedback control for ship's automatic berthing by recurrent neural network, *IFAC-PapersOnLine* **52**(21), 91 (2019), 12th IFAC Conference on Control Applications in Marine Systems, Robotics, and Vehicles CAMS 2019
- [13] N. Mizuno, Y. Uchida, T. Okazaki, Quasi real-time optimal control scheme for automatic berthing, *IFAC-PapersOnLine* **48**(16), 305 (2015)
- [14] A.B. Martinsen, G. Bitar, A.M. Lekkas, S. Gros, Optimization-based automatic docking and

- berthing of asvs using exteroceptive sensors: Theory and experiments, *IEEE Access* **8**, 204974 (2020)
- [15] A. Maki, N. Sakamoto, Y. Akimoto, H. Nishikawa, N. Umeda, Application of optimal control theory based on the evolution strategy (CMA-ES) to automatic berthing, *Journal of Marine Science and Technology* **25**(1), 221 (2020)
- [16] Y. Akimoto, Y. Miyauchi, A. Maki, Saddle point optimization with approximate minimization oracle and its application to robust berthing control, *ACM Trans. Evol. Learn. Optim.* **2**(1) (2022)
- [17] R. Sawada, K. Hirata, Y. Kitagawa, E. Saito, M. Ueno, K. Tanizawa, J. Fukuto, Path following algorithm application to automatic berthing control, *Journal of Marine Science and Technology* **26**, 541 (2021)
- [18] S. Han, Y. Wang, L. Wang, H. He, Automatic berthing for an underactuated unmanned surface vehicle: A real-time motion planning approach, *Ocean Engineering* **235**, 109352 (2021)
- [19] Z. Piao, C. Guo, S. Sun, Research into the automatic berthing of underactuated unmanned ships under wind loads based on experiment and numerical analysis, *Journal of Marine Science and Engineering* **7**(9) (2019)
- [20] S. Shimizu, K. Nishihara, Y. Miyauchi, K. Wakita, R. Suyama, A. Maki, S. Shirakawa, Automatic berthing using supervised learning and reinforcement learning, *Ocean Engineering* **265**, 112553 (2022)
- [21] Y. Liu, N.k. Im, Q. Zhang, G. Zhu, Adaptive auto-berthing control of underactuated vessel based on barrier lyapunov function, *Journal of Marine Science and Engineering* **10**(2) (2022)
- [22] A.H. Monahan, The probability distribution of sea surface wind speeds. part I: Theory and seawinds observations, *Journal of Climate* **19**(4), 497 (2006)
- [23] J. Seguro, T. Lambert, Modern estimation of the parameters of the weibull wind speed distribution for wind energy analysis, *Journal of Wind Engineering and Industrial Aerodynamics* **85**(1), 75 (2000)
- [24] I. Van der Hoven, Power spectrum of horizontal wind speed in the frequency range from 0.0007 to 900 cycles per hour, *Journal of Atmospheric Sciences* **14**(2), 160 (1957)
- [25] C. Nichita, D. Luca, B. Dakyo, E. Ceanga, Large band simulation of the wind speed for real time wind turbine simulators, *IEEE Transactions on Energy Conversion* **17**(4), 523 (2002)
- [26] L. Dostal, M. Hollm, E. Kreuzer, Study on the behavior of weakly nonlinear water waves in the presence of random wind forcing, *Nonlinear Dynamics* **99**, 2319 (2020)
- [27] A. Maki, Y. Maruyama, L. Dostal, M. Sakai, R. Sawada, K. Sasa, N. Umeda, Practical method for evaluating wind influence on autonomous ship operations, *Journal of Marine Science and Technology* **27**(4), 1302 (2022)
- [28] 牧敦生, 丸山湧生, Leo Dostal, 酒井政宏, 澤田涼平, 笹健児, 梅田直哉, 自動運航性能評価のための不規則風外乱の高速計算アルゴリズムの開発, 日本船舶海洋工学会講演会論文集 (33), 65 (2021)
- [29] 牧敦生, 丸山湧生, Leo Dostal, 笹健児, 澤田涼平, 脇田康希, 酒井政宏, 梅田直哉, 自動運航性能評価のための不規則風外乱の高速計算アルゴリズムの開発(第2報), 日本船舶海洋工学会講演会論文集 (34), 13 (2022)

-
- [30] S. Miyoshi, T. Ioki, Development of maneuvering system for realizing autonomous ships, *Class NK Technical Journal* (3) (2021)
 - [31] S. Inoue, H. Mori, Development of automated ship operation technologies, *Class NK Technical Journal* (3) (2021)
 - [32] Y.Y. Kim, K.J. Choi, H. Chung, S. Han, P.S. Lee, A ship-to-ship automatic docking system for ocean cargo transfer, *Journal of Marine Science and Technology* **19**, 360 (2014)
 - [33] S. Kohlbrecher, J. Meyer, O. von Stryk, U. Klingauf, A flexible and scalable slam system with full 3d motion estimation. in *Proc. IEEE International Symposium on Safety, Security and Rescue Robotics (SSRR)* (IEEE, 2011)
 - [34] J. Zhang, S. Singh, LOAM: lidar odometry and mapping in real-time. in *Robotics: Science and Systems X, University of California, Berkeley, USA, July 12-16, 2014*, ed. by D. Fox, L.E. Kavraki, H. Kurniawati (2014)
 - [35] E. Nelson. Berkeley localization and mapping. (2016). URL <https://github.com/erik-nelson/blam>
 - [36] T. Shan, B. Englot, D. Meyers, W. Wang, C. Ratti, D. Rus, LIO-SAM: Tightly-coupled lidar inertial odometry via smoothing and mapping. in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (2020), pp. 5135–5142
 - [37] B. Zhou, Y. He, K. Qian, X. Ma, X. Li, S4-SLAM: A real-time 3D lidar SLAM system for ground/watersurface multi-scene outdoor applications, *Autonomous Robots* **45**(1), 77 (2021)

第3章

自動着桟制御の対象船とシステム

3.1 緒言

本章では、本研究で用いた実験船についてその仕様や搭載する制御システムの概要を説明する。また、今回実施する自動着桟制御は2種類の構成により実現される。一つは、自動着桟制御の基本的な構成として、経路計画と経路追従制御からなるアルゴリズムであり、もう一つは、風外乱に対して補償する2自由度制御器である。それぞれの構成の特徴についても、本章で述べる。また、本研究では、目的に応じた自動着桟システムの2つの実装例を示す。一つはPC上で動作するソフトウェアであり、もう一つは産業用のコントローラを用いた組み込みシステムとして実装される。

3.2 対象船の概要

本研究では、大型船よりも低速で風などの外乱の影響を大きく受けると予想される小型船を対象としている。また、幅広い既存運航船舶に適用するため、対象船舶の仕様として、1軸1舵を持つ基本的な構成の船舶を想定している。そこでこれらの条件を満たすものとして、本論文で実験の対象とするのは海上技術安全研究所が管理する小型実験船「神峰」である(Fig. 3.1)。神峰の主要目をTable 3.1に示す。また、神峰の構造と装備等の配置図をFig. 3.2に示す。神峰は、1軸1舵船であり、MAU型プロペラとバウスラスターを搭載している。もともとは離島居住者の移動を円滑にするための社会実験のため、海上と陸上で交通機関の乗換なしに目的地に移動できるシームレス船として開発された[1]。その後、船員負荷低減技術、水素燃料電池システムの安全性検証等の研究に用いられてきた。本研究では本船に後述する制御システムを実装・搭載し、自動着桟に関する研究を遂行した。

3.3 制御アルゴリズムの概要

3.3.1 概要

自動着桟制御のコアとなるアルゴリズムは、経路計画、経路追従の2つの部分から構成されている。制御開始位置から着桟位置までの経路は、制御シーケンスの開始時に生成される。この方法にはいくつかの利点がある。一つは、自動制御を開始する前に着桟経路を確認できることである。



Figure 3.1 Experimental ship "Shinpo"

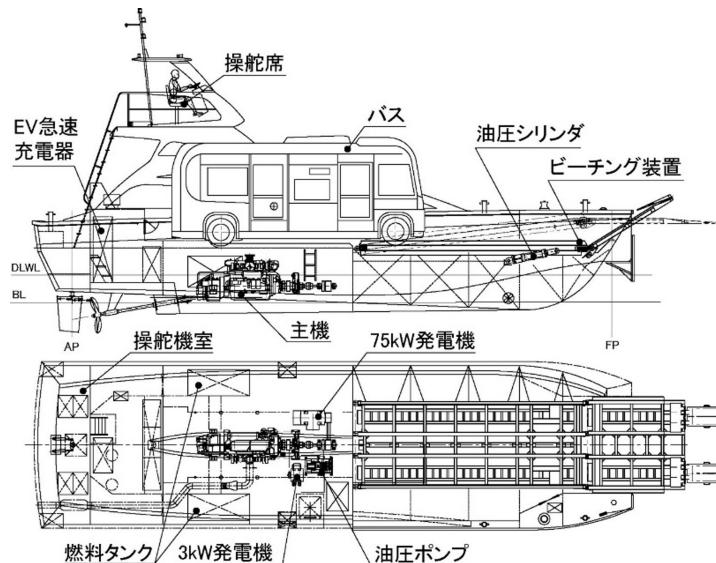


Figure 3.2 Architecture and equipment of "Shinpo"

もうひとつは、自動制御中に航路からの逸脱がわかるので、システムが現在正しく機能しているかどうかが明確になることである。さらに、このアプローチでは、自動着桟制御の問題を、着桟経路の設計、方位制御の安定性、安全な速度制御というより小さな制御問題に分割して設計することができる。

ここでは、Fig. 3.3に示すように、青の破線で示した着桟経路を4つの制御シーケンスに分割して、制御モードを切り替える。これらの切替位置は、実際に人間のオペレータが着桟する手順を想定して決定している。また、主機への過大な負荷を軽減するため、最後の停止モードを除き、プロペラ逆回転による減速を行わず、中立状態で減速することとした。減速に必要な距離を確保するため、制御モードの切り替え位置は、着桟位置から航路に沿って10, 70, 100 [m]の位置に配置されている。もちろん、他の船舶に適用する場合も、十分な減速距離を確保した上で、制御モードの変更位置を配置することが必要である。

Table 3.1 Principal particulars of the experimental ship "Shinpo"

Subjects	Value
Ship length overall, L_{oa} [m]	16.5
Ship length between perpendiculars, L_{pp} [m]	14.9
Ship breadth (1.0WL), B [m]	4.38
Ship draft, d_m [m]	0.502
Trim, τ [m]	0.348
Diameter of propeller, D [m]	1.0
Propeller pitch @ 0.7R, P [m]	0.775
Side area of rudder, A_R [m^2]	0.780
Height of rudder, H_R [m]	1.05
Mass, m [kg]	21.53×10^3
Moment of inertia, I_{zG} [$kg \cdot m^2$]	3.044×10^2
Longitudinal coordinate of center of gravity, x_G [m]	-1.260
Longitudinal projected area, A_F [m^2]	14.40
Lateral projected area, A_L [m^2]	41.52

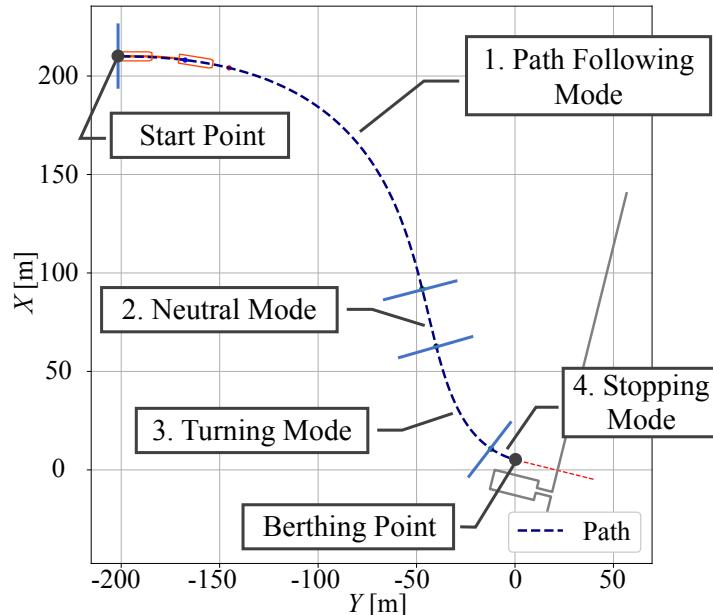


Figure 3.3 Scheme of automatic berthing control

3.3.2 経路計画

本節では、着桟経路を生成する経路計画の手法について説明する。提案する手法は、経路生成の計算量を抑え、すぐに制御を開始できる実用性を重視して設計された。

3.3.2.1 Bézier 曲線を用いた経路計画

経路計画アルゴリズムは、人間のオペレータによる実際の軌道を参考に設計されている。先行研究の結果や神峰の実運用例 [2] から、左右舷付の着桟経路としては操船開始時の船の位置から船首の方向へ伸び、桟橋に平行な角度で着桟位置へと接続する滑らかな曲線であればよいと考えられる。このような形状は、3次 Bézier 曲線で表すことができる [3]。次数 N の Bézier 曲線は、 $N+1$ の制御点 B_0, \dots, B_N で定義される N 次曲線であり、 t をパラメータとして以下のように表される。

$$\text{Bézier}(t; N) = \sum_{i=0}^N B_i J_{N,i}(t) \quad (3.1)$$

$$J_{n,i}(t) = \binom{n}{i} t^i (1-t)^{n-i} \quad (3.2)$$

ここで、 $0 < t < 1$ 、 $J_{n,i}(t)$ は次数 n の Bernstein 基底多項式であり、3次 Bézier 曲線は t の 3 次多項式として表現できる。Fig. 3.4 は、制御点 B_i ($i = 0, \dots, 3$) による Bézier 曲線による着桟経路の模式図である。Fig. 3.4において、制御点 B_0 は、船舶の初期位置に配置される。 B_1 は、 B_0B_1 の角度が船の初期方位角と等しく、その長さが $L_s = 0.6|y_{berth}|$ となる点である。 B_2 は $L_e = 80$ [m] で桟橋に平行な点、 B_3 は着桟目標位置である。ここで、着桟目標位置は、実験の安全性と、船から桟橋まで係留ロープを投げることができる距離の観点から、 $(x_{berth}, y_{berth}) = (5, 0)$ [m] とした。長さ L_s は開始位置によって変化するため、どの位置から自動着桟制御を開始しても、ほぼ同じ角度で桟橋に接近することができる。進入角度の調整は、主に長さ L_e によって行われる。他の船へ本手法を適用する場合は、長さ L_e を変えることにより桟橋へアプローチするときの角度を変えることが可能である。

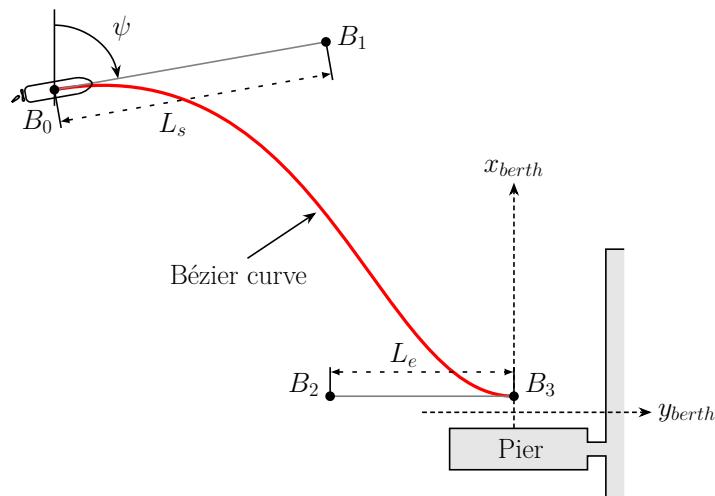


Figure 3.4 Path planning using a cubic Bézier curve with control points

3.3.2.2 助走距離を設けた着桟経路

前節で提案された、着桟経路計画は、風外乱の影響が小さい時などではうまく動作する。一方で、後述する経路追従制御の特性として、一度生じた経路偏差（CTE:cross track error）を小さくすることに時間を要するというものがある。実際に、本研究で提案する自動着桟制御手法を適用し、実船実験した結果の航跡図が Fig. 3.5 に示されている。この例では、風外乱を受ける状況で制御を行っているが、制御初期において船舶の航跡が青破線で示される計画経路から大きく逸れていることがわかる。この制御初期に発生した経路偏差は、風の向きとは異なる方向に発生しており、風外乱による偏差とは考え難い。制御初期に発生した経路偏差が小さくなるまでに 200 [m] 程度の距離を必要としている。このように着桟制御において、風外乱による影響よりも着桟初期に発生した経路偏差の影響が無視できないことがわかる。

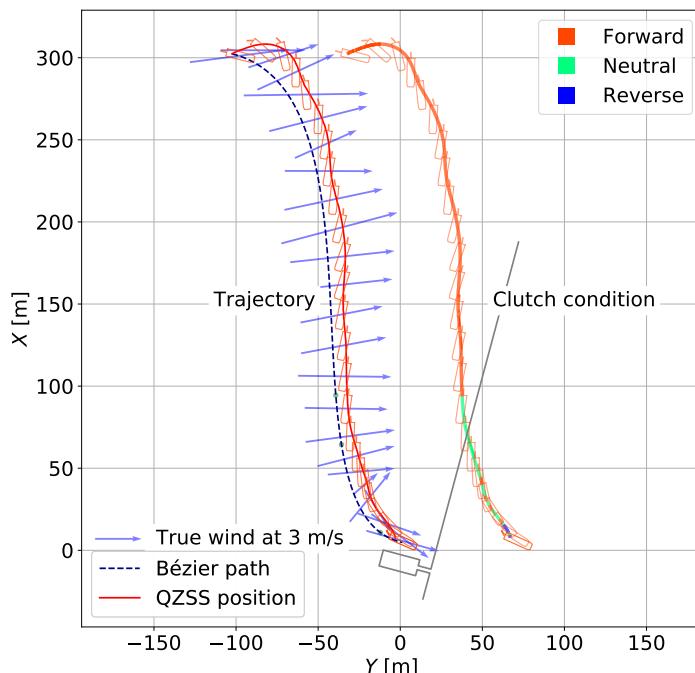


Figure 3.5 Full-scale experiment result of proposed method of automatic berthing control.

制御初期の経路追従時に船体の軌跡が外側に膨らんでしまうという問題は、トラックコントロールの状況と関連している。直線で構成されるレグから円弧で表現される曲線経路 (curved track) へ進行したとき、操舵を始めるタイミングで発生する船体運動のキックが問題になる。キックは操舵により船を回頭させようとする方向と逆に働くため、結果的に回頭が遅れることになる。この問題を回避するため、TCS のようなシステムでは、通常 WOP (Wheel Over Point) のように旋回開始位置を円弧軌道の開始位置より前に設定し、旋回動作の遅れを軽減する工夫がなされている [4]。先の経路計画手法について改めて見てみると、制御開始位置での経路の接線方向は船の初期方位角と平行であるが、経路はそのあとすぐに桟橋の方に曲がっている。つまり、船は制御開始地点で既に旋回を開始しなければならなくなるが、初期方位変化は小さいため、キックの発

生も相まって回頭角速度が発達するためには時間を見る。このため、着桟制御の初期段階において、航跡が計画経路の外側に膨らむように進むといった現象が起きる。

この問題を解決するために、図15のような曲線軌道上のTCS(Track Control System)の例のように、直線的な助走区間を設けることを提案する。Fig. 3.6は、直線的な助走区間を設けた場合の経路計画の概要を示している。助走区間の大きさをどのように取るのが適切かについては、第5章において、数値シミュレーションを用いた検証を行う。

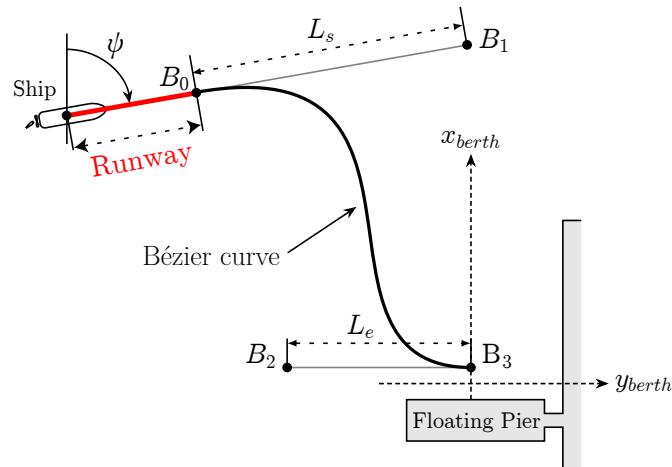


Figure 3.6 Path planning using a cubic Bézier curve with runway

3.3.3 経路追従制御

本節では、計画された着桟経路を精度良く追従し船舶を着桟位置に誘導する経路追従の制御手法について説明する。

3.3.3.1 Pure pursuit 法に基づく経路追従制御

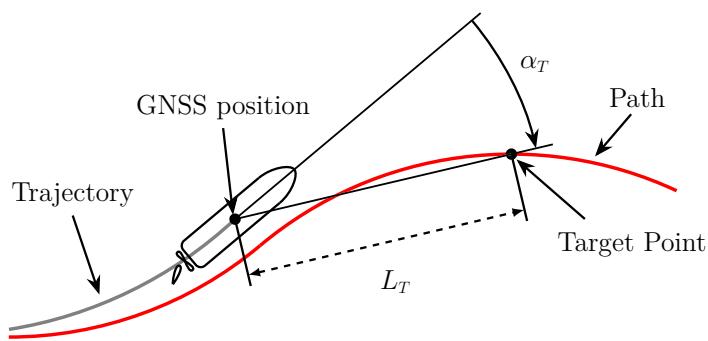


Figure 3.7 Path following control using pure pursuit and autopilot (PPA)

本研究では経路追従制御アルゴリズムとして、ロボティクスや車の自動運転[5]などの分野で広く使われる経路追従アルゴリズムであるpure pursuit[6]を採用した。Pure pursuitは、Fig. 3.7の

ように予め設定された方向を持つ有向な経路に対して、一定距離だけ前方の経路上の点を目標点として、これに到達するように旋回制御を行うアルゴリズムである。ここで、経路上の船体位置（ここではミッドシップとする）との最接近点と目標点までの距離 L_T を前方注視距離（Look ahead distance）と呼ぶ。船体位置は、後述する GNSS により取得されるものとする。ここで、GNSS とは、Global Navigation Satellite System の略称であり、全地球測位システム（GPS）、準天頂衛星システム（QZSS）等の略称である。Pure pursuit 法を船舶に適用する場合、目標点の方位角を操舵により追従する必要があるため、目標方位角に対する PD 制御の HCS を組み合わせることで、目標点までの角度偏差をもとに舵角を算出する。PD ヘディングコントローラを用いて、目標点までの方位角度の差 α_T に基づき算出される舵角を δ^* とし、最大で ± 45 [deg] の範囲に収まるよう舵を切る。実際の計算式は次のようになる：

$$\delta^* = \text{clip}(K_P[\alpha_T]_{-180^\circ, 180^\circ} - K_D r, -45^\circ, 45^\circ) \quad (3.3)$$

ここで、 K_P と K_D はそれぞれ比例ゲインと微分ゲインである。ゲインパラメータ K_P と K_D は、多くの船舶に搭載されている HCS（Heading Control System）の設定に基づき決定することができる。いくつか事前に実施したシミュレーションと実船計測の結果に基づき、 $K_P = 3.0$, $K_D = 1.0$ とした。参考までに、通常航行時の K_P の HCS 設定は神峰の場合、2.0 程度である。HCS のゲイン設計では、pure pursuit が経路の先を見て制御することから微分制御的な働きをするため、外乱に対するロバスト性と経路追従性能を改善することができる。ただし、通常の保針制御が不安定にならない範囲 [7] で経路上の最も曲率が大きい区間の追従性を考慮して K_P を大きな値に設定する必要がある。設計の目安として、経路上の最も曲率の大きい区間の追従性を考慮することが望ましい。今回、前方注視距離 L_T はシミュレーションと実船実験による円軌道の追従制御の結果から、 L_{pp} の 1.6 倍である 26.4 [m] とした。前方注視距離 L_T は基本的に固定であるが、外乱の影響により船体位置と経路上の最接近距離が設定した前方注視距離より大きくなる場合は、経路上の最も近い点を目標点として使用する。また経路計画について、桟橋に近づいたときに着桟位置から先の目標点を計算する必要があるため、Fig. 3.4 の点 B_3 から桟橋に平行になるように適当な距離だけ直線経路を延長する。

3.3.3.2 速度制御

本研究で使用した実験船の制御システムは、主機テレグラフ指令値とクラッチを操作して船速を制御することができる。ただし、主機への機械的負荷への対策や実船実験での安全性を考慮し、今回は、主機回転数をアイドリング回転数に固定して着桟制御を行うこととした。そのため、クラッチの切り替えのみで船速を制御することになる。クラッチは正転、中立、逆転の 3 つの状態を取ることができる。具体的な制御則として、Fig. 3.3 に示された 4 つの制御部において、次のように船速を制御する：

- (1) 経路追従モードでは、クラッチを前進に設定する。
- (2) ニュートラルモードでは、クラッチを中立にし、強い向かい風などで surge 速度が $u < 1.0$ [m/s] になったときの場合に前進に切り替える。
- (3) 回頭モードでは速度を 1.0 [m/s] 以下に抑えるために原則的にクラッチをニュートラルに

入れるが、次の3つの条件のいずれかが満たされたとき、クラッチは前進に入れる：

- (A) $\alpha_T < 0$ [deg]かつ surge 速度 $u \leq 1.0$ [m/s] のとき。
 - (B) GNSS から推定された船体のミッドシップ位置から着桟位置までの直線距離が 50 [m] 以上かつ $|\alpha_T| > 5$ [deg] のとき。
 - (C) $u < 0.3$ [m/s] のとき。
- (4) 停止モードでは、クラッチを段階的に逆転させながら、surge 速度 u が 0.1 [m/s] 以下となるようにする。まず、 -10.0 [m] $< x_{berth} < -1.0$ [m] かつ $u > 0.5$ [m/s] のとき、クラッチをリバースに切り換える。加えて、桟橋までの距離に関係なく $u > 0.1$ [m/s] のときはクラッチを逆転に入る。着桟位置の手前の $x_{berth} < 0$ [m] で $u < 0$ [m/s] の場合(つまり着桟位置手前で船速が負になった場合)、クラッチは前進に切り替わる。これらの条件を満たさない場合、クラッチを中立にする。

本研究では、右舷付けで着桟することを想定している。一方、左舷付けでの着桟の場合は、(3)の(A)の不等号の方向が逆になる。本制御則では、停止モード以外では減速時にクラッチをリバースにすることはない。これは、クラッチの正逆切り替えによる主機関の過負荷を防止するためである。したがってニュートラル航行による減速を行う回頭モードでは、目標方位角と船首方位角の差が大きくなても船速が十分小さい状態でなければプロペラ後流を舵に当てるによる急旋回を行わない設計とした。

3.3.3.3 風外乱補償制御

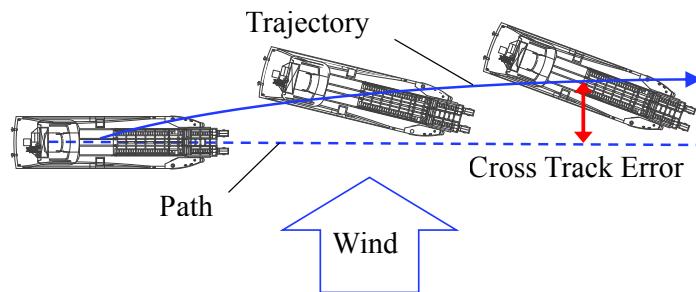


Figure 3.8 Wind disturbance affecting path following control.

Table 3.2 Settings for estimation of coefficients

Subjects	Forward	Neutral	Reverse
u [m/s]	2.072	2.072	0.0
v [m/s]	0.0	0.0	0.0
r [deg/s]	0.0	0.0	0.0
n_P [rps]	3.09	0.0	-3.09

本節で説明する風外乱補償制御は、前節で説明した経路追従アルゴリズムに風外乱を補償するフィードフォワード制御機構を加えたものである。Pure pursuit 法は、目標地点からやや前方を目

指すように船首方位角を制御することで経路追従制御を行う。この方位制御を行うのが PD（比例微分）制御ベースの方位制御システムである。この制御系の利点は、必要な制御パラメータが少ないとわかる。特に、pure pursuit 法は、船の位置から目標点までの距離である前方注視距離 L_T を調整するだけで、設計することができる。

風外乱が経路追従制御に与える影響として、経路からの偏差が生じる流れを Fig. 3.8 に示す。この図に示す通り、特に船体の横方向からの風を受けると、船体位置の進路からの逸脱が大きくなることがわかる。風による進路偏差を低減することで、進路追従制御の精度を向上させることができる。そこで本研究では、風圧力の横成分をキャンセルするための操舵量を決定するフィードフォワード制御系を示す。そこで船の運動モデルに基づいて、船体に働く風圧力を考える。一般的な MMG モデルに基づく運動モデルは船体のミッドシップを座標原点として次のような形で表現される。

$$\left. \begin{array}{l} (m + m_x)\dot{u} - (m + m_y)vr - x_Gmr^2 = X \\ (m + m_y)\dot{v} + (m + m_x)ur + x_Gmr\dot{r} = Y \\ (I_{zG} + x_G^2m + J_z)\dot{r} + x_Gm(\dot{v} + ur) = N_m \end{array} \right\}. \quad (3.4)$$

ここでは議論を簡単にするために、重心に船体固定座標系の原点を置いた場合の船の操縦運動の方程式を次のように考える：

$$\left. \begin{array}{l} (m + m_x)\dot{u} = (m + m_y)vr + X(u, v, r, \delta) \\ (m + m_y)\dot{v} = -(m + m_x)ur + Y(u, v, r, \delta) \\ (I_{zG} + J_z)\dot{r} = N(u, v, r, \delta) \end{array} \right\}, \quad (3.5)$$

ここで、 X, Y, N は重心周りの船体に作用する外力とモーメントを表す。この運動方程式には、船体に作用する力やモーメントなど、多くの非線形項が含まれている。この方程式から、制御のためのダイナミクスモデルが、次のように導かれる [8, 9]：

$$\begin{aligned} (3.5) &\Rightarrow \begin{bmatrix} m + m_x & 0 & 0 \\ 0 & m + m_y & 0 \\ 0 & 0 & I_{zG} + J_z \end{bmatrix} \begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} X_l \\ -(m + m_x)ur + Y_l \\ N_l \end{bmatrix} \\ &\Rightarrow \begin{bmatrix} m + m_x & 0 & 0 \\ 0 & m + m_y & 0 \\ 0 & 0 & I_{zG} + J_z \end{bmatrix} \begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{r} \end{bmatrix} \\ &= \begin{bmatrix} X_u & 0 & 0 \\ 0 & Y_v & Y_r - (m + m_x)u \\ 0 & N_v & N_r \end{bmatrix} \begin{bmatrix} u \\ v \\ r \end{bmatrix} + \begin{bmatrix} T_x & D_x \\ 0 & D_y \\ 0 & D_n \end{bmatrix} \begin{bmatrix} n_P \\ \delta \end{bmatrix} + \begin{bmatrix} X_A \\ Y_A \\ N_A \end{bmatrix}, \end{aligned} \quad (3.6)$$

ここで、 X, Y, N は重心位置の周囲で船体に作用する線形化された外力とモーメントを表す。 X_u, Y_v, Y_r, N_v, N_r は流体力学的な微分である。Eq. (3.6) の最後の項は、プロペラと舵による力とモーメントを直線化したものである。 T_x, D_x, D_y, D_n の項は、線形化された力とモーメントの係数である。Eq. (3.6) の Sway の運動方程式は以下の通りである：

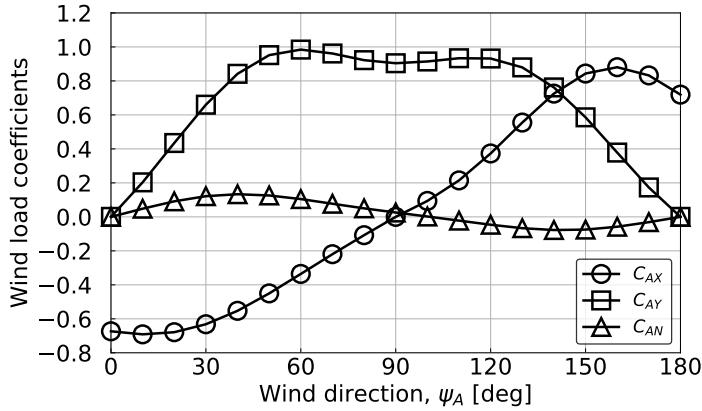


Figure 3.9 Estimated wind load coefficients.

$$\dot{v} = \frac{1}{(m + m_y)} [Y_r v + \{Y_r - (m + m_y)u\} + D_y \delta]. \quad (3.6')$$

ここで、舵力の横成分 F_{dy} を $F_{dy} := D_y \delta$ と定義する。最後に、フィードフォワード制御のための線形化された横舵力モデルを以下のように導出する：

$$F_{dy}/(m + m_y) = D'_y \delta_{FF}, \quad (3.7)$$

ここで、 D'_y は、舵によって引き起こされる横力の線形化モデルの正規化係数である。添え字の FF は、フィードフォワード制御則による計算した舵角であることを示す。この制御則を前提に、前節の数学モデルを用いて、操舵によって船体に作用する横力を解析した。解析対象は、Table 3.2 に示す正転、中立、逆転の各状態に対応する運動である。正転、中立については、主機アイドリング回転数で直進しているときの運動状態をモデル化したものである。逆転については、クラッチが停止モードでのみ逆転に入るため、船速がゼロの状態で解析している。解析結果を Fig. 3.10 および Table 3.3 に示す。ここで、 D_y は Eq. (3.7) を満たすように計算された係数であり、Fig. 3.10 の結果において舵角が 0 [deg] の値で代表している。船体に作用する横方向の風圧力は、前節で述べたように次のようなモデルで表される：

$$Y_A = \frac{1}{2} \rho_A A_F U_A^2 C_{AY} (\psi_A) \quad (3.8)$$

Fig. 3.9 に実験船の風圧力係数を示す。その結果、Eq. (3.7) と Eq. (3.8) から $Y_A + F_{dy} = 0$ を解くと横風荷重を打ち消す舵角が求まることがわかる。得られた舵角に基づいて操舵することで、風圧力の横成分をキャンセルすることができるが、あくまで推定であり、旋回方向の風荷重によるモーメントや潮流などの外乱要因や、センサ値の誤差が発生する可能性がある。これらの要因による誤差は、フィードバック機構を持つ PPA アルゴリズムによって一括して補正される。舵角指令 δ^* は、風力補償フィードフォワード制御で算出した δ_{FF} と PPA アルゴリズムによるフィード

ドバック制御で算出した舵角 δ_{PPA} の和であり、以下のようなになる：

$$\begin{aligned}\delta^* &= \delta_{PPA} + \delta_{FF} \\ &= K_P \alpha_T - K_D r - \frac{m + m_y}{2D'_y} \rho_A A_F U_A^2 C_{AY}(\psi_A),\end{aligned}\quad (3.9)$$

ここで、 K_P と K_D はオートパイロットの比例ゲインと微分ゲインである、 r は目標角度の誤差、 r は旋回速度である。これが、feedforward, pure pursuit and autopilot (FFPPA) アルゴリズムである。これは 2 自由度 (2-DoF) コントローラである。舵角と船体に作用する風圧力の関係を 1 次式で表すことで計算量を抑え、従来手法のリアルタイム性を維持しつつ、風外乱に対する制御の高い制御を行うことができる。本研究では、 $K_P = 3.0$, $K_D = 1.0$, 前方注視距離は $1.6L_{pp}$ と設定した。

Table 3.3 Estimated coefficients of linear lateral rudder force model

Subjects	Forward	Neutral	Reverse
D'_y	-0.003272	-0.002771	0.0

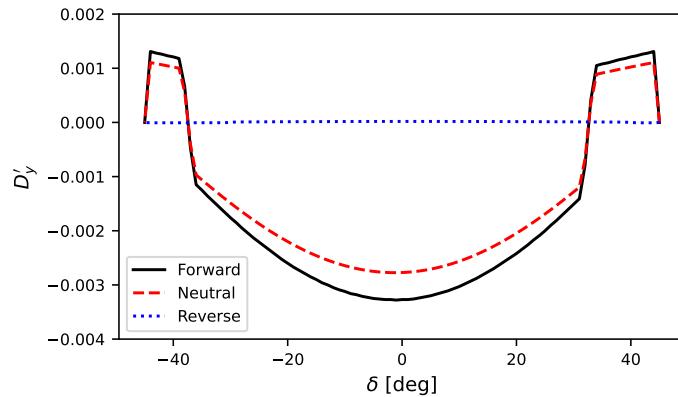
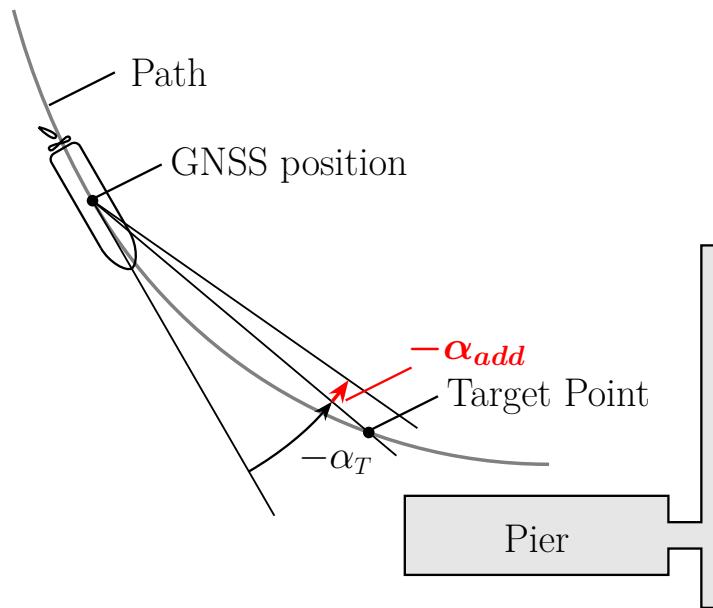


Figure 3.10 Estimated coefficients of lateral rudder force.

3.3.4 経路追従制御の特性と補正角の導入

Pure pursuit 法による経路追従制御の特性として、計画偏差 (cross track error) が大きくなるに従って、経路に垂直に針路を取るようになる。これは Pure pursuit による経路追従機構は前方注視距離は一定であるため、外乱などによって経路からの偏差が大きくなると相対的に、経路方向に対しての経路上の船の最接近点と船体位置を結ぶ方向の角度が大きくなるためである。結果的に経路上で前進する方向よりも経路へ復帰する度合いを高めることになり、この作用によって外乱を受けたときの経路偏差をある程度軽減することができる。

特に制御終盤において船と桟橋の距離が小さい場合、船首を桟橋に向けたまま接近することは、乗組員に恐怖感を与える可能性がある。このようなアプローチでは、船首が桟橋に衝突する危険性も高くなる。したがって安全のために、桟橋に接近する際の方位角の変化を適切に制御する

Figure 3.11 Geometric meaning of α_{add} on approaching pier.

ことが重要である。なお、本船にはバウスラスターが搭載されているが、連続運転時のバウスラスターへの過負荷が懸念されるため、本研究ではバウスラスターを使用しないことにした。また、バウスラスターを使用した際のヨーモーメントは、船に前進速度がある場合には小さい。そこで、桟橋への進入時の方方位角は、旋回モードと停止モードにおける純粋な追従による指令舵角を算出する際に、以下のように α_T に補正角 α_{add} を加算することで調整することとした。

$$\delta^* = \text{clip}(K_P[\alpha_T + \alpha_{add}]_{-180^\circ, 180^\circ} - K_D r, -45^\circ, 45^\circ) \quad (3.10)$$

Fig. 3.11に桟橋進入時の方方位制御における α_{add} の幾何学的な意味を示す。 α_{add} を加えることで、元のアルゴリズムより早く旋回を開始することができる。 α_{add} を大きくすることで、着桟位置からの距離が小さい場合に制御機に入力される α_T を大きくすることができる。ここでは、桟橋への接近時の方方位角を調整するために、補正角 α_{add} を導入している。一方、着桟時に適切な方位角となるように Bézier 曲線の経路そのものを再設計する方法もある。後者の方がシンプルではあるが、この場合、軌道が必ずしもパスに沿うとは限らない。その上、旋回モード以降の極低速状態では制御余裕がほとんどないため、キックやドリフトを利用した非常に複雑な制御を行い、方位角を制御することになる。この方法で最終的に着桟位置まで到達することは可能であるが、正しく経路を追従できているか否かをシステムの操作者が画面上の経路偏差の情報から判断することが難しくなるという欠点がある。そこで、実用性の観点から、補正角度を加算する方式を採用した。Eq. 3.10を用いて α_{add} を変化させながら自動着桟を行った実船の試験結果は、第5章で示される。

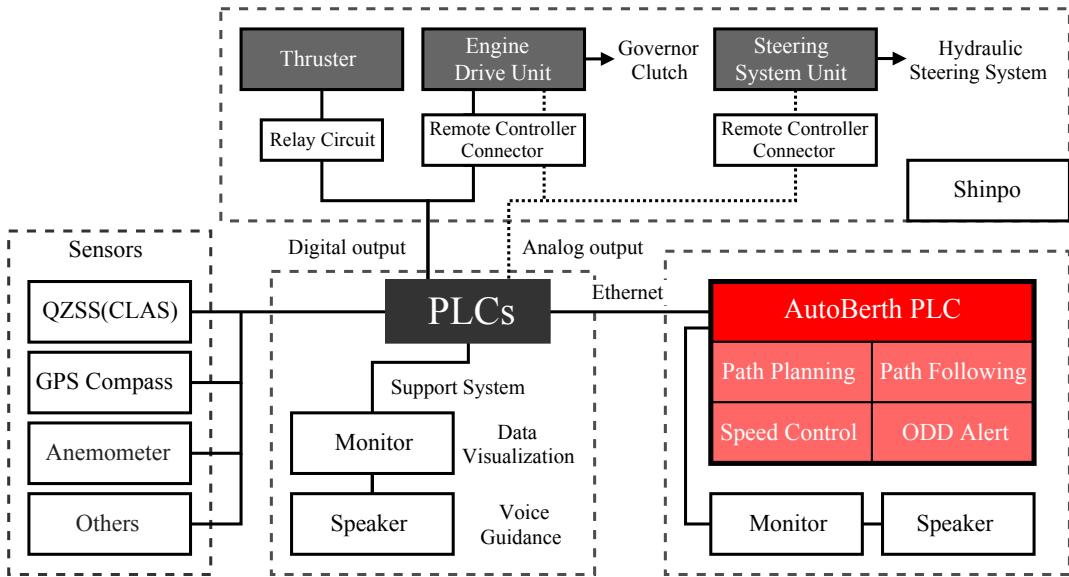


Figure 3.12 Architecture of onboard control system and AutoBerth PLC

3.4 実験船の制御システムの構成

自動着桟制御を行うにあたり、神峰には油圧操舵システムや主機の制御システムとして PLC (Programmable Logic Controller, シーケンサとも) を中心としたシステムを開発し搭載している [10]. PLC は組み込みのリアルタイムコントローラであり、豊富な IO を有しているため複数の機器とアナログ/デジタル信号でやりとりができるほか、ロギングや自己の状態監視等の機能を有している。

小型実験船「神峰」には、Fig. 3.12 に示される制御システムの模式図では、中心にある PLC から船の各種アクチュエータが接続され、また他にセンサ類、着桟操船支援用モニタ、また制御計算を行うパーソナルコンピュータ (PC) もしくは別の専用 PLC が接続される。この図では、自動着桟制御用 PLC システムである AutoBerth PLC を接続した際の関係を示している。本制御システムでは、信頼性と拡張性に優れたプログラマブルロジックコントローラ (PLC) が船内の様々なアクチュエータの制御機能を担っている。油圧ステアリングシステムと主機のリモートコントロールシステムは、船内の PLC 制御システムで監視・操作することができ、舵、主機ガバナー (エンジンテレグラフ指令に対応)、クラッチ、バウスラスターを操作することができる。また、計測されたセンサー値はすべて船内の PLC 制御システムに集約、記録され、制御システムに接続されたタッチディスプレイに内容が表示される。船の制御は、PC から送られるコマンドを PLC が受け取ることで各種アクチュエータの制御を行う。同時に神峰に搭載された各種センサー情報が PLC を介して PC もしくは別の制御用 PLC に送られる仕組みとなっている。



Figure 3.13 Sensors installed on "Shinpo". QZSS Receiver (TOP), Anemometer, GPS compass, antenna for QZSS (LOW)

3.5 センサー

着桟制御のため、「神峰」には様々なセンサーが設置されている。神峰の通常の着桟操船では、甲板員が係留ロープを安定して桟橋へ渡るために、船側と桟橋の距離が最大でも 5 [m] 程度になるように船を誘導する必要がある。一方、操船法の教科書 ([11, 12]) によれば、桟橋に対して 10~15 度の浅い角度で接近する場合、潮流や風の影響が十分に小さいときには、係留前に船幅 B の 1.0~1.5 倍程度の距離で停止させるとある。ここで、本船は L_{OA} が 16.5 [m] であるため、水線での船の形状を $l_{pp} \times B$ の矩形と仮定すると、ミッドシップ位置を船体位置として着桟制御を行った場合、桟橋に対して 10 度の浅い方位角で理想的に桟橋に近づいたとしても、船首側の隅と桟橋の距離は約 1.43 [m] まで接近する。実海域の着桟制御においては、外乱やセンサの誤差による位置制御の誤差が発生することを考慮すると、桟橋と船首が接触しないような安全な距離を保つためには、十分に高い測位精度が必要である。これらのことから、実験で安全な着桟制御を行うためには、1 [m] 以下の精度で船位を取得できることが望ましいと考えられる。そこで、船舶

の測位には、CLAS (Centimeter Location Augmentation Service) 対応の受信機 Chronosphere-L6 を採用した。CLAS とは、QZSS が送信するセンチメートルレベルの補強情報である。受信機の感度を上げるために、アンテナは操舵室の屋根の上に設置されている。さらに、方位角を得るために GPS コンパス Furuno SC-50 を搭載し、また、相対風速と風向きを計測する風向風速計も操舵室の上に設置されている。このほか、プロペラ回転数、エンジン回転数、舵角、ロール、ピッチなども計測できる。これらのセンサは PLC システムへと接続され、自己位置、船首方位などの船体の動的情報をリアルタイムに計測、記録、画面表示と音声によるモニタリングが可能となっている。また、今回は搭載される風向風速計のデータを用いて後述する風外乱補償制御の計算に利用する。Fig.3.13 は、実験船に搭載されたみちびきの受信機および、操舵席の屋根の上に置かれたセンサ類の様子を示している。さらに、LiDAR を用いたセンサ系の冗長化についても、検討した。その詳細は次章で述べる。

本研究では、自動着桟アルゴリズムは、筆者らの研究 [13] ではノートパソコンで動作するプログラムとして実装されており、船内制御システムに接続することが可能である。さらに、安定的な動作が可能な PLC への移植も行った。次節以降で、PC および PLC で動作する自動着桟機能についてそれぞれ説明する。また、本研究では、まず、「AutoBerth PLC」の代わりにノートパソコン上で動作する「AutoBerth」を使用し、新たに提案したアルゴリズムの検証を行った。

3.6 自動着桟制御システム

3.6.1 パーソナルコンピュータによる制御

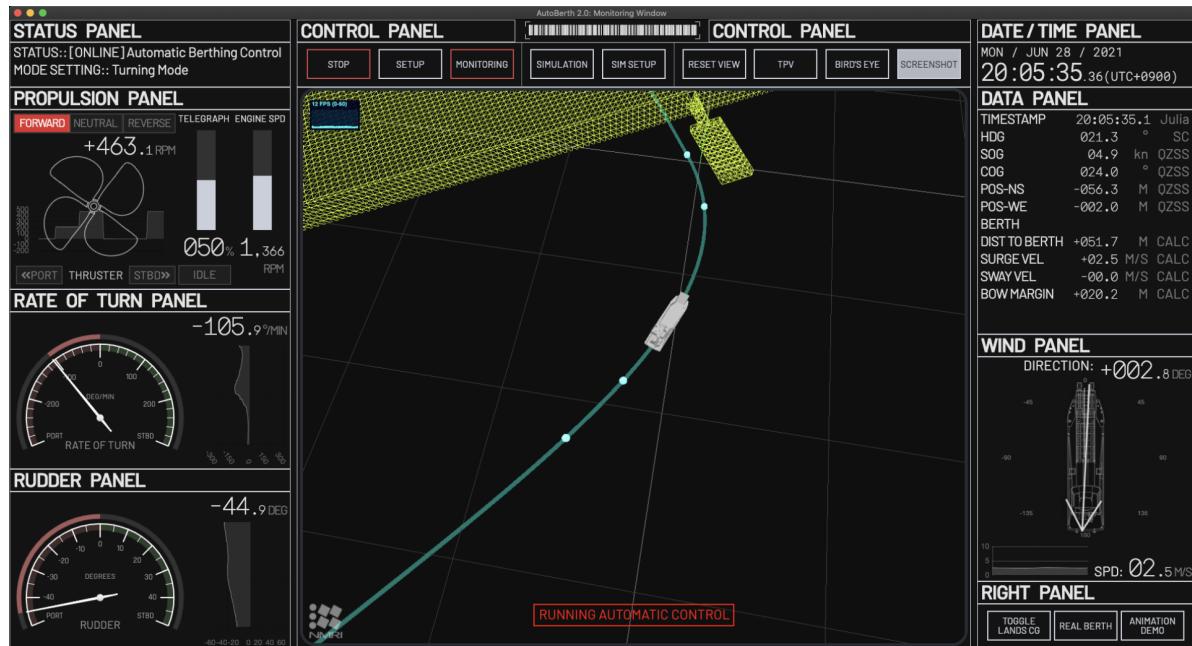


Figure 3.14 Screenshot of the AutoBerth UI

AutoBerth は、PC (パーソナルコンピュータ) 上で動作する自動着桟ソフトウェアである。AutoBerth は、センサーデータの解析、自動着桟のためのアクチュエータ制御の計算に加えて、自動

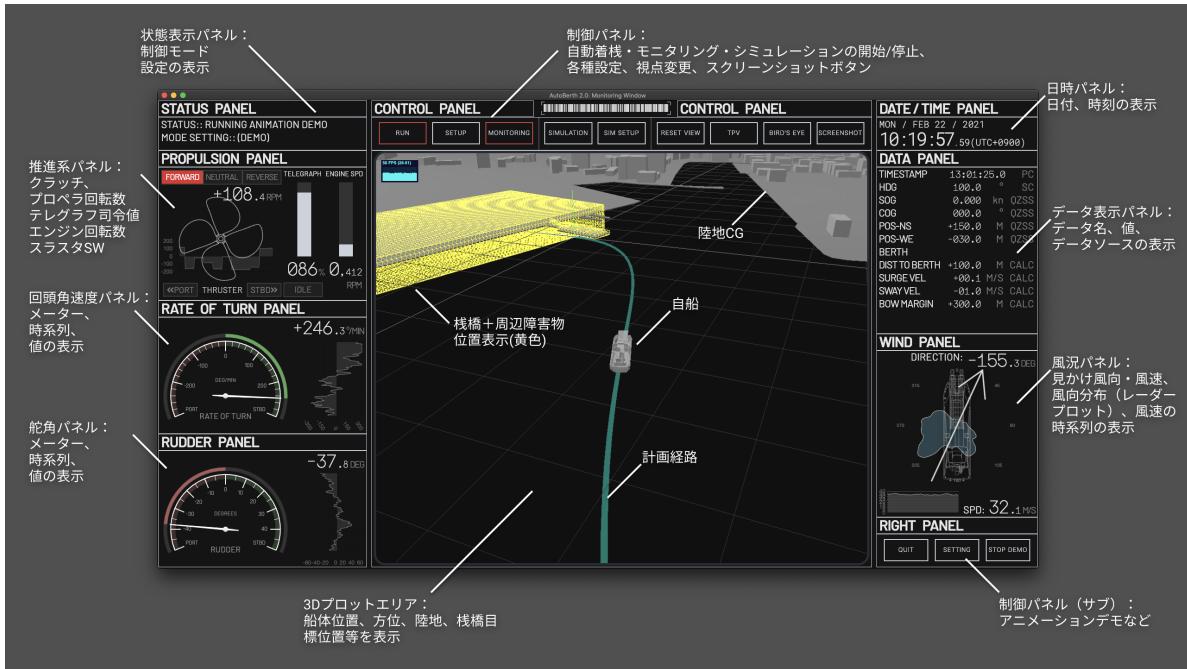


Figure 3.15 Components on the AutoBerth UI

制御中の船舶の運動状態や自然環境等を一目で確認できるように Web ベースの GUI を有している。AutoBerth の GUI を Fig. 3.14 に示す。また、画面上のコンポーネントの説明を、Fig. 3.15 に示す。画面上では、センサから取得された各アクチュエータの状態が左列に、運動状態や外乱等の運航に関する情報が画面右側の列に表示されている。特に左列のアクチュエータは、自動着桟制御中に、司令されたコマンドを受けて船が動作しているかを一瞥して確認できるように、数字による表示だけでなく、グラフやメータ等によるグラフィカルな表示をするとともに、瞬時値のみならず時系列データも表示することで状況を分かりやすくユーザに表示する。同様の工夫は、風況パネルに取り入れられており、数字のみならず船体に対する見かけ風向・風速をグラフィカルに表示する他、レーダープロットの形で、風向・風速の履歴を積み上げグラフとして表示し、過去の風況を含めた情報を提供する。中央には、船体、着桟の計画経路、着桟目標（桟橋および周辺岸壁）とその他の陸地が 3DCG で表示される。AutoBerth は、画面上のボタンを操作することで自動着桟制御を任意のタイミングで開始できる他、実験船の自動化システム（PLC）から特定の文字列を受け取ることでリモートから自動着桟制御を開始・停止する事ができる。自動着桟制御が動作している時以外は、モニタリングだけの機能を提供し現在の船舶・環境の情報を表示することができる。このような画面表示機能は、現在の自動制御が適切に動作しているかを操作者が監視する上で重要である。着桟中に、センサ異常や想定以上の外乱の発生等があった場合に、制御計算はそのまま動いてしまっているといった状況など意図しない状況であるか否かといったことは、単に舵や船体位置だけを見ても判断がつかないため、経路に対する船体運動や、外乱情報、制御モード等の情報を多角的にかつ瞬時に確認して判断する必要がある。このような目的の下、本機能の GUI は設計されている。

AutoBerth と船上のシステムとの関係は、Fig. 3.16 に示している。AutoBerth は GUI には 3 D を含めたリッチな表現に適した Web ベースの技術を用いてる。UI の実装は主に Electron 上で動

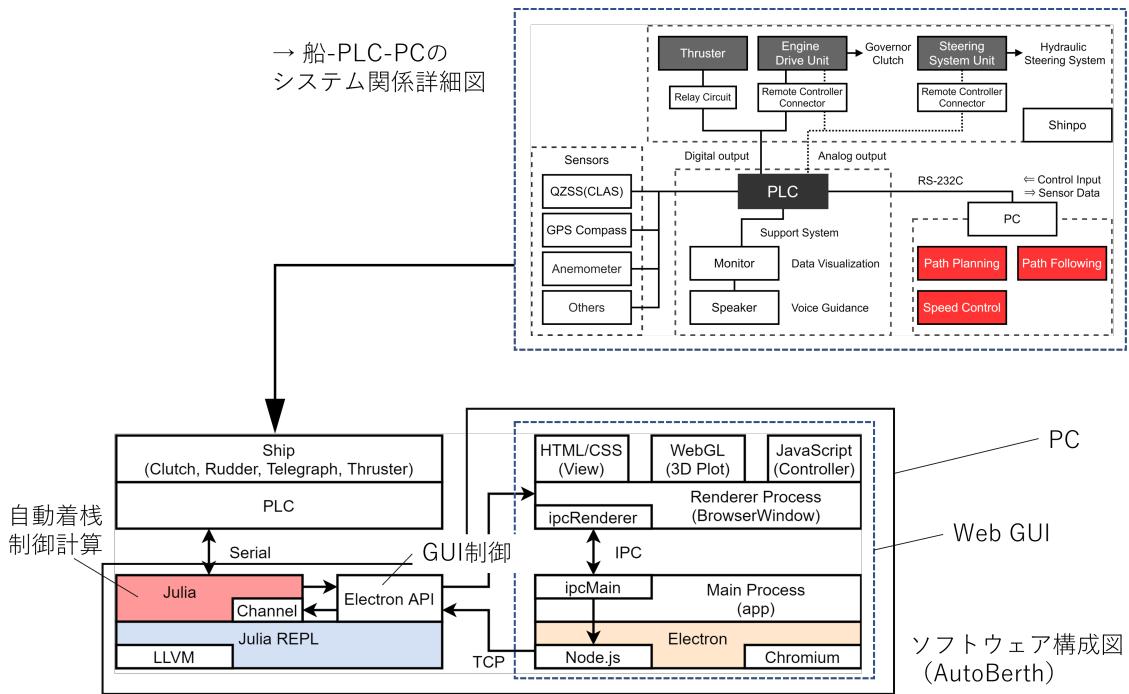


Figure 3.16 System architecture of the AutoBerth UI

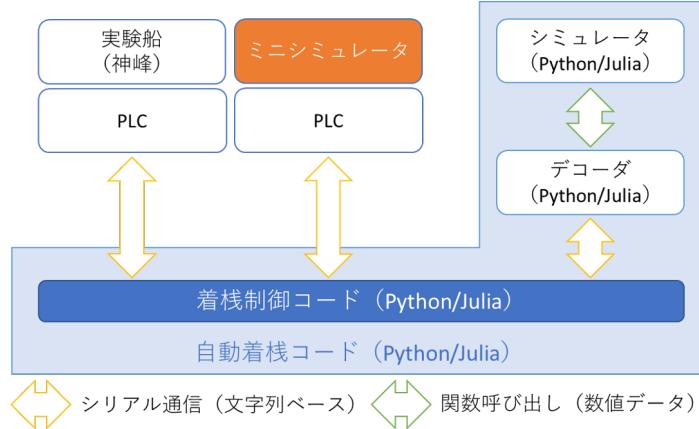


Figure 3.17 Development environment for prototyping of automatic berthing system

作する JavaScript により実現されている。AutoBerth の主たるプロセスは、Julia 言語 [14] により実装されたコードにより実行され、GUI (Electron API) とのプロセス間通信も担っている。Julia で実装されたコードは、船上の PLC とシリアル通信を介して互いに接続されている。Julia 言語は強力な型推論と LLVM という JIT (Just in Time) コンパイラを導入することで動的型付け言語でありながら、その他インタプリタ方式の言語と比較して高い計算速度を誇るために、経路計画や経路追従制御等の計算についても処理される。

自動着棧を含む自動運航船の機能を開発する上で、一般に実船を使った検証に使える時間は少ない。特に、就航している商用船へ搭載するシステムの場合は、修繕のためにドック入りしたタイミングなどに集中して開発やシステムの調整を実施する必要があり、試験の工数にも制限がかか

る。したがって、実船搭載前にシミュレーション等を用いて陸上における開発フローに重点を置く必要がある。そこで、より効率的な開発を行うための開発環境の整備についても注力した。Fig. 3.17は、今回整備した開発環境の概要を示している。まず、図中株および右側を囲む青枠の部分はPC上で完結するシミュレーション環境を表している。ここでは、開発対象となる自動着桟コードと同じ動作環境で使用できるシミュレーションコードを実装している。シミュレーション環境と制御コードは、実船のPLCシステムの通信を模擬した機構として、やり取りする文字列データのデコーダを通じて通信を行う。この機構により、PC上で開発したコードを一切の改変なく、そのまま実船のPLCシステムに接続して動作させることが可能となり、実運用の際に必要なコード修正を最小限にすることでバグの発生を減らす効果が見込める。シミュレーションコードは、制御コードと密接に連携し、高速に計算結果を提供することができる。この段階では、制御パラメータのあたりをつけたり、制御コードの基本的な実装のバグを解消したりする作業を効率的に実行することができる。一方で、この段階では検証が同一PC上の同一実行環境において検証されるのみであるため、ソフトウェア上のバグのみが検証されることになる。実際には、PLCシステムとRS-232Cを用いたシリアル通信を行うため、PLCシステムとの通信や、PLCシステムに実装されているインターロック等の安全機構との干渉といったハードウェアに起因する問題を検証する必要がある。このような問題は、仮に不具合があったときにソフトウェアと比較してバグを解消することが難しい場合がある。したがってこの作業も、実船搭載前に検証できることを望ましい。提案の開発環境の図中央にある海技研が保有するミニシミュレータは、訓練用のシミュレータを改造したもので、実船に搭載されるPLCと同じ構成のシステムを接続して、模擬的に船を制御することのできる環境を提供する。このミニシミュレータとPLCの環境を用いて、基本的な動作を確認することができ、ソフトウェアのハードウェアに起因する不具合を事前に解消することができる。ここまで工程を経て陸上で可能な限り検証を行ったソフトウェアを実船に搭載し、実際の運動性能に合わせた最終的なパラメータチューニングを実施する。実船搭載時に、ソフトウェアやハードウェアの不具合についても事前にある程度の検証が済んでいるため、意図しない動作や事故が発生する可能性を減じることができる。このように、陸上で可能な範囲の検証を事前に済ませておくことで、実船に搭載可能なシステムの開発を効率的に実施することができる。

一方で、AutoBerthは、PC上で動作する関係上、実験船に搭載されているPLCシステムの機能をすべて使うことはできない。基本的には、PLCから提供されるセンサデータと、PCからPLCに送信される制御コマンドの文字列のみが相互にやり取りされるのみである。PLCが有するウォッチドッグタイマ等の自己監視機能は、PCは有しておらず、AutoBerthが動作中にソフトウェアがなんらかの原因でフリーズしたり通信が途絶することを監視することが難しい。ウォッチドッグタイマとは、監視対象のプログラムとは別に定期的に起動するプログラムであり、監視対象のプログラムがハングアップするなどしていないかを定期的にチェックし、異常があれば例外処理を投げる。プログラムが止まったときだけでなく、例えば制御コードの計算や、画面描画のための計算に時間を要し、特に制御コマンドが規定時間以内に出力できていない場合は、プログラムとしては正常に動いているが、要求された時間周期で制御できていないため、これも検知する必要がある。PCでこれらの機能を実現しようとした際には、ウォッチドッグタイマを別のプロセスとして走らせ、プログラムの動作を監視する必要があり実装の難易度が高い。また、ウォッチドッグタイマのプロセスも巻き込まれて、同時にハングアップする場合など、PCがハングア

ップした場合に適切に監視ができないことも懸念される。一方で、PC 上のプログラム開発は人が理解しやすい高級言語による記述が可能であり、ライブラリ等を用いるなどして高速な開発・試験・評価のループを回すことが可能である。そのため開発の大部分は、PC 上のプログラムで動作を確認することが効率的である。一方で、ある程度機能が成熟してきた段階で、安定的な動作や自己監視等の安全機能を盛り込む目的で、PLC への実装は一つの有用な選択肢である。次節では、AutoBerth の機能を移植し、安全機能を盛り込んだ AutoBerth PLC の実装について述べる。

3.6.2 PLC による制御

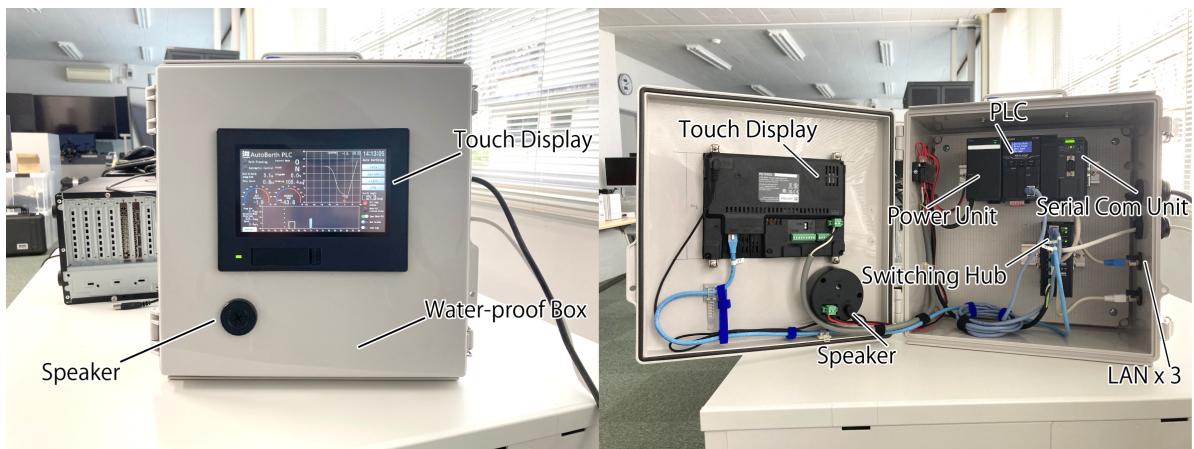


Figure 3.18 Automatic berthing (AutoBerth) PLC system

本節では、着桟操船のための自動化システムの実用化例として PLC への実装例を示す。自動着桟制御をより実用的な実装に落とし込むために、PC 上の実装にはいくつかの問題があった。例えば、PC では、動いているソフトウェアの監視のためのウォッチドッグタイマの実装が難しかったり、割り込み処理やタイマ管理といった低レベルの制御機能の実装が難しいという課題がある。PC のこのような制約を解決できるハードウェアとして、PLC がある。PLC は産業用コンピュータであり、一般的に工場設備などでリアルタイムコントローラとして使用される。一般的な商船では、マイレージモニターやメインスイッチなど、船内でのセンサーデータの通信やデータ管理、解析のための処理装置として使用されている。実験船では、センサーデータの収集、解析、ロギング、着桟操船やウェイポイント航行の音声サポート、制御などの船内制御システムにおいて PLC が使用されている。そこで、本研究では前節で提案した自動着桟アルゴリズムを PLC に自動化システムとして実装し、このシステムを AutoBerth PLC と名付けた。PC 上で動作する AutoBerth ソフトウェアは、実験船の PLC システムからの制御を可能にする機能を実装したが、PLC にシステムを実装することで、さらに高度な機能として、自動着桟システムと本船の自動化システムの状態を相互に監視するといった機能を提供することができる。PLC は、産業用のコントローラとして優れた安定性とリアルタイム性を有しており、自動制御のコントローラに適している。Fig. 3.18 に、本研究で実装した AutoBerth PLC システムを示す。PLC は防水ケース内に収められ、ケースに取り付けられたタッチディスプレイでセンサ等の情報の表示や音声ガイダンス、アラートの発報を行う機能を有している。ケースを防水仕様にするため、電源ケーブルやイーサネットケ

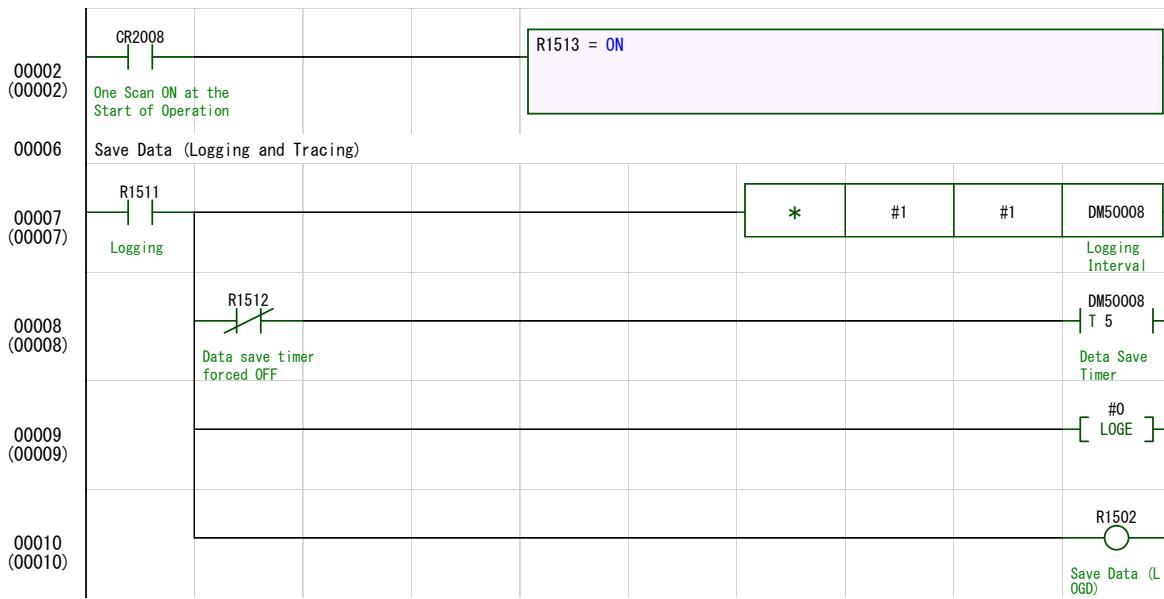


Figure 3.19 Example of the rudder diagram in AutoBerth PLC (translated by the authors). In the diagram, "R" is a relay, "DM" is a data memory, "T" is a timer and "CR" is a special relay named the control relay. For example, lines 00007 to 00010 will be executed when the relay "R1511" is ON in this program.

ケーブルは防水のケーブルランドを通してケースの外に配線している。一般的な PC 上を走る OS とは異なり、PLC 上で動作するプログラムは IEC61131-3 で定義される 5 つのプログラミング言語セットを用いて開発される。まず PLC のプログラマは、基本的にラダー図 (Ladder Diagram) を使ってシーケンス制御処理を記述する。Fig. 3.19 は基本的なラダー図の例を示している。ラダー図は、リレー回路やタイマー回路等をプログラム上で再現し、シーケンス制御を実現する。実際の AutoBerth PLC のラダー図はその一部は、Fig. 3.20 のようになっている。提案システムの基本機能である自己保持回路、タイマー処理、アラームのトリガーは、ラダー図を用いて実装した。このラダー図では、さらに拡張された機能として PC で使用する高級言語に似たスクリプトを記述することができる。自動着棧制御コードの IF-ELSE-THEN 句などの関数や条件実行を含む複雑な処理は、構造化テキスト (ST) 言語で実装されている。本提案システムの経路計画や経路追従制御等の計算は、この ST 言語によって実装されたものである。AutoBerth PLC と船内制御システムのシステムアーキテクチャは、すでに前出の Fig. 3.12 に示されている。

AutoBerth PLC は、船内制御システムから受信したセンサデータをわかりやすく表示する機能も有している。実際のシステム GUI の画面は Fig. 3.21 の通りである。AutoBerth と同様に、船体位置を俯瞰する図や各センサー値等が一目で分かるように実装されている。またユーザビリティを考慮するため、日英表示や日中・夜間での動作に対応したライトモード・ダークモードといった複数の UI を実装している。

さらに、PC 上で動作する AutoBerth になかった機能として、ODD(Operational Design Domain)に関する警告、緊急停止、状態監視のためのオプションメニューも備えている (Fig. 3.22)。ODD とは、事前に設計された自動化システムが適切に動作することのできる、自然環境や航行環境等

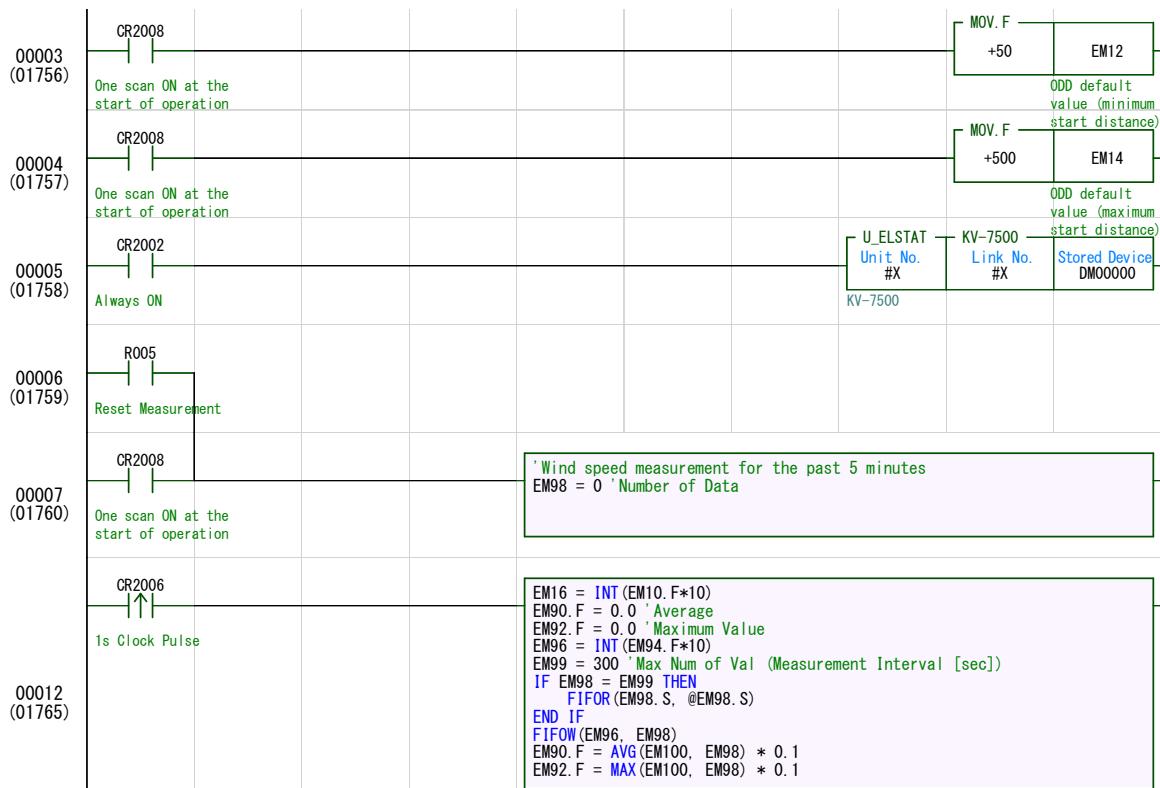


Figure 3.20 A part of the ladder diagram of the AutoBerth PLC.

の条件を定めたものである。一般的に、MASS の自動化システムは安全性を担保するために、その動作を保証することのできる特定の ODD に基づいて設計されている必要がある。具体的には、システムが ODD 内にあるかどうかを監視する機能、またはシステムが ODD から外れた場合に警告と乗組員へと操作権を移譲するヒューマンフォールバックレスポンスを行う機能を持つ必要がある。フォールバックレスポンスという語には、システムがシステムの故障状態を検出した場合に、通常稼働の機器を用いて、設計時に定義された MRC(Minimum Risk Condition) に自動的に遷移する MRM (Minimum Risk Maneuver) も含まれる。このような機能は、無人船、遠隔操作船、あるいはブリッジに誰もいない有人船(B0 船)などの自動化レベルで使用されるものである。本研究では、有人船でシステムが支援的に自動制御を行う状況を想定し、システムによる注意喚起や船上オペレーターへの引継ぎ等のフォールバック対応を行うための注意喚起機能を実装した。自動着棧中は、人間の乗組員が常にシステムの状態を監視し、システムが ODD から外れたときや、システムに何らかの異常が発生したときに適切に対応できるようにアラートを出すようなシステムとして設計することとした。ODD の対象となる条件の中には、風況など、運航中に急激に変化するものがある。このような条件では、例えば突風などの一時的かつ船体運動に対して影響が限られるような状況へ過度に反応しないように、一定の測定周期を設定して平均値等を計測するなどして、瞬時値の急激変化に影響されないようなシステム設計が必要である。このような点を考慮しつつ、本研究では制御開始位置と風速に対して ODD を設定する機能を実装した。この機能では、経路計画を適切に実行できるように制御開始時に制御開始位置があらかじめ設定された範囲内にあること、5 分間の平均風速と最大真風速があらかじめ設定された基準値以下で

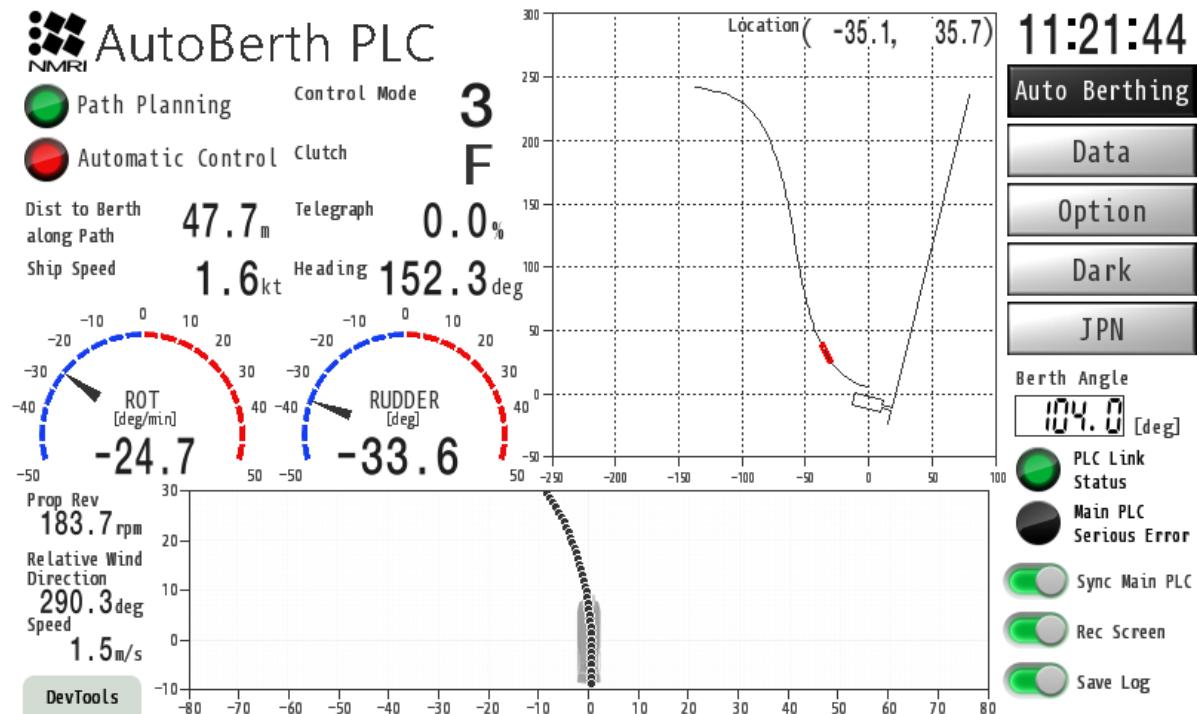


Figure 3.21 Main graphical user interface of AutoBerth PLC

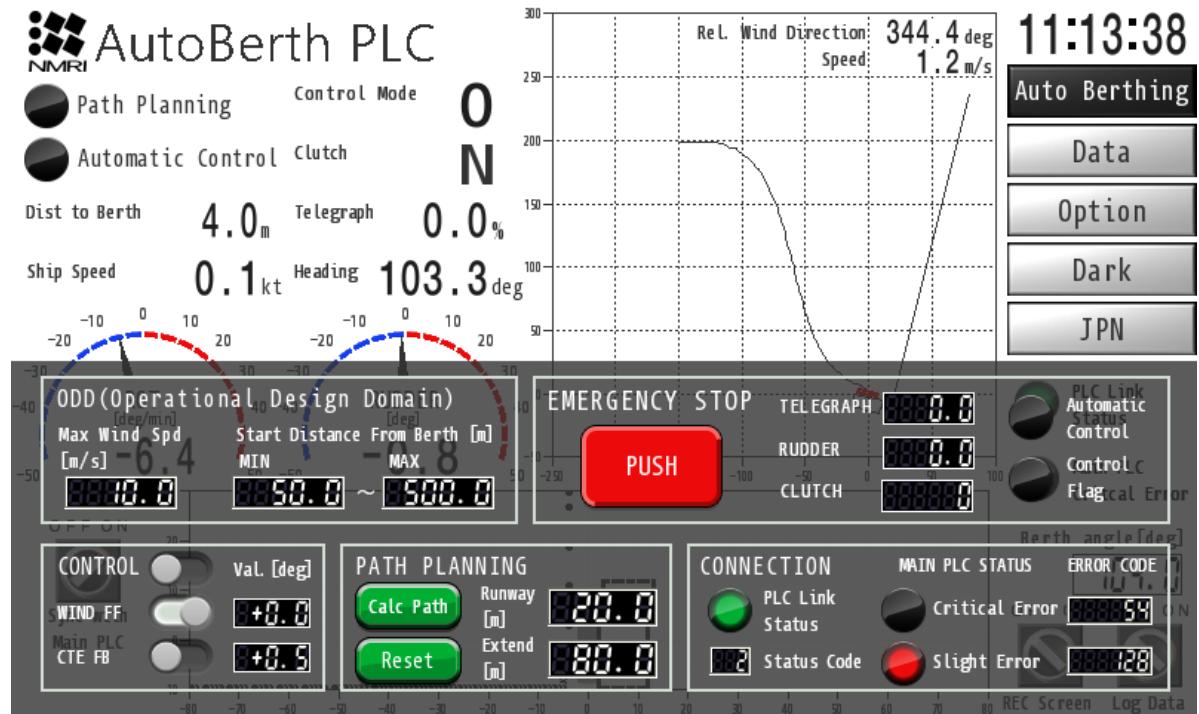


Figure 3.22 Option menu of AutoBerth PLC

あることを保証する。また、PLC の重要な機能のひとつに、自己診断機能がある。これは PLC の状態を常に監視し、PLC の状態を専用メモリにフラグの形で保存するものである。提案システムでは、船上の自動化システムを構成する主要な PLC と AutoBerth PLC が、Ethernet を介して通信し、相互監視を行う。ここでは、お互いのシステムの通信状態を監視する機能と、互いのシステムが持っているそれぞれの自己診断結果を相互に監視し、通信切断時や相手のシステムで処理のエラーや計算ループの計算時間が既定値を超えるなどのエラーが発生した場合に警告を発する機能を実装した。異常状態は、ゼロ除算等の計算エラーが含まれる軽微なエラーと、システムが継続的な動作ができなくなる重大エラーに分類される。特に、システムが使えなくなるような重大エラーが発生した場合に、音声とアラームによって乗組員に警告される。これに関連して、自動航行中に Ethernet ケーブルが切断された場合、音声ガイダンスやアラーム等で乗組員に警告することで、通信状態の異常を知らせ、その後の対応を促す機能も実装されている。AutoBerth PLC のヒューマンフォールバックレスポンスのための警報機能は、実際に実験船上で適切に動作することを確認した。また、MRM の実装例として、通信が遮断された場合に船舶を緊急停止する機能を、船上の PLC システムに実装した。実際に、Ethernet ケーブルの接続を故意に切断し、AutoBerth PLC との通信を遮断したときの MRM の発動テストの結果を Fig. 3.23 に示す。この試験では、通信が遮断されたタイミングで非常停止シーケンス制御のリレーが ON になり、主機回転数を上げてただちに船速を落とすことが確認できた。

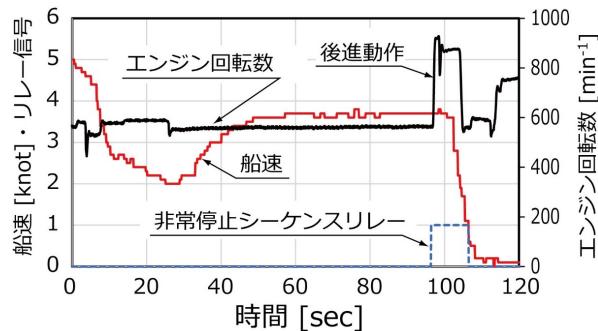


Figure 3.23 Test result of detection of communication disconnection and minimum risk maneuver.

3.7 結論

本章では、自動着桟を実施するにあたり、対象船と対象船の制御システムについて示した。また、本研究において提案する自動着桟アルゴリズムについても述べた。提案するアルゴリズムは、経路計画と経路追従制御から構成されており、それぞれの機能を別々に設計できるだけでなく、制御実行前の着桟経路の確認や、ODD の確認といった機能を提供する上でも都合がよい。さらに、船体に働く風外乱については、フィードフォワード制御による 2 自由度制御アルゴリズムを提案した。アルゴリズムの導出においては、3 自由度の操縦運動数学モデルを状態方程式で定式化し、風圧力と線形化した舵力の関係式を導出した。また、安全性の観点から、緊急時に乗組員が停止できるようにアプローチ角度を補正する手法についても述べた。さらに、対象船の制御システムとセンサ系についても示した。センサ系の冗長化を目的とした LiDAR によるセンシング技術に

関する研究については次章で述べる。最後に、自動着桟システムについて、2つの実装例を示した。1つ目はPC上で動作するAutoBerthであり、Webベースの技術を用いたリッチなGUIと、Julia言語による高速な制御計算が特徴である。このシステムの利点は、その開発フローにもあり、船上での開発時間を最小化すべく、陸上におけるSILS(Software in the Loop), HILS(Hardware in the Loop)環境を示した。2つ目の実装は、PLCへ機能を移植したAutoBerth PLCである。PLCに移植するにあたり、ラダー図やST言語により自動着桟制御の機能を実現した。また、PCでは実装の難しい状態監視機能を有し、設定されたODDに基づくアラート機能、船上システムとの相互状態監視、MRMとしての通信途絶時の緊急停止機能を実装した。これらの機能は、安全な自動運航船のシステムを構築する上で重要な要素であり、本研究において具体的な実装例を示した。

第3章の参考文献

- [1] 平田宏一, 宮崎恵子, 斎田賢次郎, 荒谷太郎, 松倉洋史, 吉村健志, 西崎ちひろ, 三宅里奈, 田村兼吉, 離島の交通支援のためのシームレス小型船システムの開発, 海上技術安全研究所報告 **16**(4) (2016)
- [2] E. Saito, M. Numano, K. Miyazaki, K. Hirata, Maneuvring motion simulation to support berthing operation of small crafts - proposal of a berthing operation support system, International Maritime and Port Technology and Development Conference and International Conference on Maritime Autonomous Surface Ships (2019)
- [3] M.P. Kamermans. A primer on Bézier curves. <https://pomax.github.io/bezierinfo/> (2018). (2023年5月31日閲覧)
- [4] 羽根冬希, 《第13回》船舶用オートパイロットの制御技術, 計測と制御 **51**(11), 1086 (2012)
- [5] B. Paden, M. Áp, S.Z. Yong, D. Yershov, E. Frazzoli, A survey of motion planning and control techniques for self-driving urban vehicles, IEEE Transactions on Intelligent Vehicles **1**(1), 33 (2016)
- [6] G. Heredia, A. Ollero, Stability of autonomous vehicle path tracking with pure delays in the control loop, Advanced Robotics **21**(1-2), 23 (2007)
- [7] 元良誠三, 船体運動力学 (電子訂正版) (2007)
- [8] T. Fossen, *Guidance and Control of Ocean Vehicles* (Wiley, 1994)
- [9] E.I. Sarda, H. Qu, I.R. Bertaska, K.D. von Ellenrieder, Station-keeping control of an unmanned surface vehicle exposed to current and wind disturbances, Ocean Engineering **127**, 305 (2016)
- [10] 澤田涼平, 平田宏一, 北川泰士, 齊藤詠子, 宮崎恵子, 上野道雄, 福戸淳司, 着桟操船の音声支援と自動着桟制御に関する研究, 海上技術安全研究所報告別冊 **20** (2019)
- [11] 岩井聰, 操船論 (海文堂出版, 1977)
- [12] 井上欣三, 操船の理論と実際 (成山堂書店, 2011)
- [13] R. Sawada, K. Hirata, Y. Kitagawa, E. Saito, M. Ueno, K. Tanizawa, J. Fukuto, Path following algorithm application to automatic berthing control, Journal of Marine Science and Technology **26**(2), 541 (2021)
- [14] J. Bezanson, S. Karpinski, V.B. Shah, A. Edelman, Julia: A fast dynamic language for technical computing, CoRR **abs/1209.5145** (2012)

第 4 章

自動着桟を想定した LiDAR の適用

4.1 緒言

本章では、制御以外の自動着桟技術として、LiDAR を利用した研究成果を示す。本章がカバーする範囲として、1. 航行環境、特に岸壁を考慮した経路計画への適用、2. 点群地図作成と自己位置推定、3. 岸壁検知の 3 つの結果を示す。これらは、より実用的な自動運航船を実現するために、制御以外の部分で必要な機能であり、より安全で冗長性のあるシステムの構築を目的として検討された。

各論に入る前に、LiDAR センサの概要を説明する。本研究で使用する LiDAR は、センサから照射されるレーザ光の反射時間 (ToF: Time of flight) を計測することで距離を測るセンサである。今回使用するのは、Velodyne 社の VLP-32C である (Fig. 4.1)。VLP-32C は、32 ラインの LiDAR センサであり全周 360 度を計測することができる 3D LiDAR の一つである。Fig. 4.2 は、船に載せた LiDAR で計測した点群データを表示している様子である。このように、LiDAR を用いて自船の周辺の情報を取得する事ができる。



Figure 4.1 LiDAR sensor: VLP-32C.

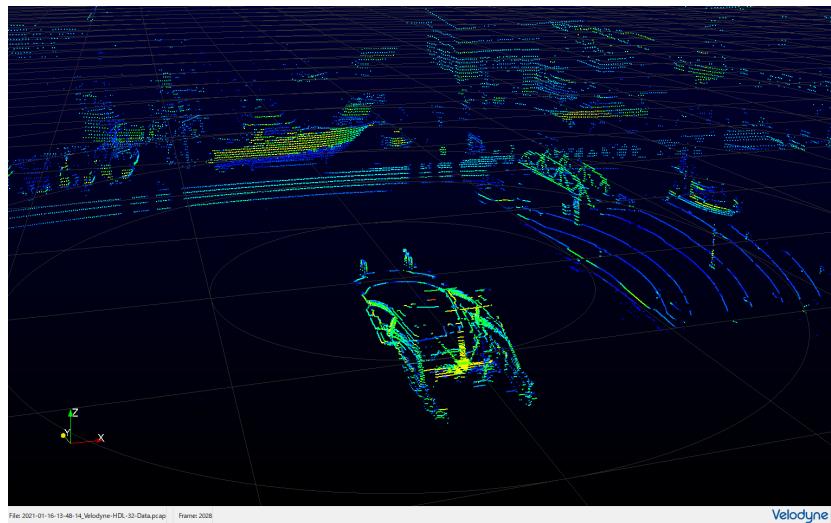


Figure 4.2 Example of point clouds obtained with LiDAR.

4.2 岸壁等を考慮した経路計画への適用

これまでに述べてきた Bézier 曲線を用いた経路計画手法は、計算量が小さく、また経路形状の調整を 4 つの制御点で直感的に行えることが利点であった。一方で、実用的な観点から航行環境、例えば岸壁等の障害物を避ける安全な経路生成ができないという課題がある。これまでには、ODD を設定することで経路が適切に生成できる範囲からでのみ利用できるように制限を設けていたが、理想的には周囲状況を考慮した経路が生成されることが望まれる。OOD の制限を減らし、適用可能な海域を広げるためには、海域の情報をもとに航行可能な領域の中で追従しやすい経路を生成することが必要である。そこで本研究では、経路探索アルゴリズムのひとつである Hybrid-state A* アルゴリズムを適用し、航行環境を考慮した着桟制御に適切な経路を生成する手法について述べる。障害物の情報として、実験船「神峰」に設置した LiDAR により計測した点群データを用いて計算を行った結果について示す。

4.2.1 Hybrid-state A* アルゴリズム

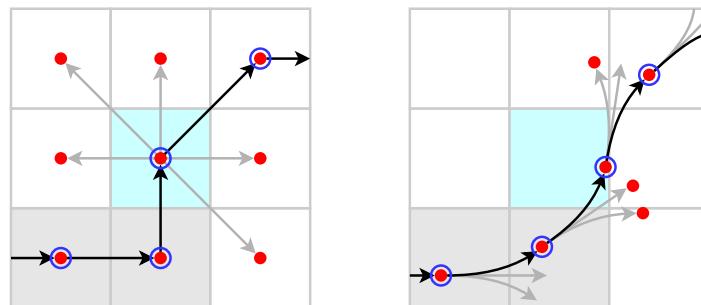


Figure 4.3 Comparison Search algorithms: classic A* (discrete state, left) and hybrid-state A* (continuous state, right).

Hybrid-state A* アルゴリズム（もしくは Hybrid A* とも）は、A* と呼ばれる古典的なアルゴリズムをベースとした経路探索のためのアルゴリズムである [1]。Hybrid-state A* の探索方法について、古典的な A* アルゴリズムとの比較図を Fig. 4.3 に示す。まず古典的な A* は、まず探索領域をグリッド状に分割し、ゴールまでの経路を逐次的に探索する手法である。A* はグラフ探索アルゴリズムの一つであり、初期位置を表すスタートノードから、障害物に専有されていない隣接するグリッドに対応するノードを順番に調べていき、最終的にゴールのノードまでの経路を探索する。探索では、同時に各ノードに設定されたコストが計算される。スタートからゴールまでの最短経路を探索する場合は、各ノードのコストはそのノードに至るまでの移動距離と効率的な探索を促進するためのヒューリスティック関数の値が対応する。よくあるヒューリスティック関数は、各グリッドからゴールまでの距離を返す単純なものである。このコストの値が小さくなる方向にノードを探索することでゴールまでの目的の経路を見つけるのが古典的な A* アルゴリズムである。Hybrid-state A* は次に述べる 3 つの工夫がなされている。1 つ目に探索の際に、4 次元の探索空間 (x, y, θ, r) を使って探索する。ここで x, y は元の探索領域における位置座標、 θ は船首方位角、 r は経路に対する移動方向で 0（前進）と 1（後進）の 2 値で表される。もともとの 2 次元平面から 4 次元空間での探索にすることで、Hybrid-state A* の探索は、2 次元空間を連続的に探索することができ、得られる軌跡は A* で得られるようなグリッドの中心をつないだ離散的な経路ではなく、連続的な曲線で表現される。一方で、探索する次元が増えることで増大する計算時間を軽減しより効率的に探索をおこなうために、動的計画法を用いたコストマップをヒューリスティック関数として用いる。これにより、湾内などの入り組んだ地形であっても、袋小路になる経路を探索することを回避してゴールまでの経路を効率的に探索できる。3 つ目の工夫として Analytic Expansion と呼ばれる手法を導入されている。Analytic Expansion は、ノードの探索によって求められた途中までの経路から、ゴールまでを曲線で接続するようにして経路を拡張することで連続値で指定される終端状態を満足する経路を生成する。通常、この経路の拡張においては、Reeds-Shepp パス [2] と呼ばれる経路生成アルゴリズムが用いられる。Reeds-Shepp パスは与えられた 2 点の位置と向きから、円弧と直線で構成される有向な最短経路を生成するアルゴリズムである。ここまでステップを経て得られた経路の候補は、衝突判定を行って有効な経路かを判定する。衝突判定は、移動体の形状を表現する長方形領域に障害物の点が含まれるかどうかを計算することで行う。この流れを繰り返して最終的に衝突のない有効な経路が見つかった時点で終了となる。実際に、因島マリーナ周辺の海岸や浅瀬を境界とする障害物領域を点群として作成し、Hybrid-state A* を用いて経路探索を行った計算例を Fig. 4.4 に示す。地図の上に重畠されたカラーマップは動的計画法を用いて計算されたコストマップを表している。桟橋に近いほどコストが小さくなるが、一方で障害物を挟んだ位置では桟橋との直線距離が小さくても桟橋までは障害物を回り込んでいく必要があるため、総じてコストが大きくなっている。得られた経路は、スタート位置から浅瀬の北側あたりまでは Hybrid-state A* のノード探索により得られた経路で、そこから桟橋までを構成する円弧と直線で構成される経路は Reeds-Shepp パスとなっている。Hybrid-state A* の探索により拡張された総ノード数は、874 個であった。

4.2.2 LiDAR による計測データを用いた例

神峰の着桟操船時に甲板上に設置した LiDAR (Velodyne VLP-32C) で計測したデータのうち、ある時刻 1 フレーム分（全周 360 度）のデータを抜き出し、センサから 20 [m] 以内の点と各点群

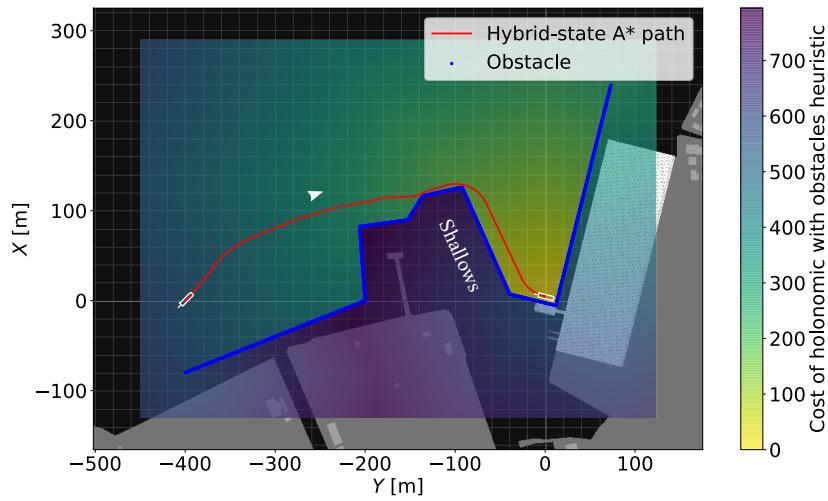


Figure 4.4 Path planning using Hybrid-state A* for berthing control in waterway geometry defined as obstacles.

の LiDAR の強度 (intensity) の値が 25 以下をノイズとして除外したデータを使用した。強度の低いデータを除いたのは、航行中の自船の曳き波や海面に反射した点群データを計算から除外するためである。LiDAR により計測された点群を用いて、経路を計算した例を Fig. 4.5 に示す。Fig. 4.5 では計測時の位置から経路を計算したものである。この海域では、桟橋までここまで近づくと針路を塞ぐような障害物は無いため、計算された経路は Reeds-Shepp パスのみで構成されてたものが求められている。より複雑な状況や、障害物の反射強度によっては、また異なった結果になる可能性がある。LiDAR を使った障害物検知においては、たまたまある時刻に点群が得られない場合があり、この例では、画像左にある防波堤の先端が検知されていない。実際にはこのように取り逃している部分については、移動していくうちに検知できる可能性があるが、解決策としては、よりライン数の多い LiDAR を使用するほかに、過去の数フレーム分のデータと合わせて使用するなどして、一時的に計測が漏れた部分のデータの補完を行い検知確率や精度を上げるなど対策が必要であると考えられる。

先程の例では、計測位置と桟橋までを遮る障害物がない状況での計算例だった。次に LiDAR のセンサデータに基づいて障害物との衝突を回避し、桟橋への至る経路について計算を行った。そこで、LiDAR で点群を取得した位置から船体位置を意図的に変えて経路を計算した例を Fig. 4.6 に示す。LiDAR の計測位置と船体の初期位置が異なる点を断っておく。この例では、桟橋の北側で船首方位が岸壁に向かっている状態から少し船首を振ったのち、一度青の実線で示される経路に沿って後進して船首方位を桟橋へ向けてから、赤の実線で表せる経路を前進して追従し桟橋まで至るといった結果が得られた。このときの探索した総ノード数は 121 個である。

LiDAR で計測した点群を用いた、経路計画について結果を示した。LiDAR のセンサデータを利用することで事前に障害物の情報を持っていないなくても安全な経路設計が行うことができる。一方で、一般的な LiDAR では水深の情報を得られないため、航行可能な領域の情報の取得には別の手段を用意する必要がある。

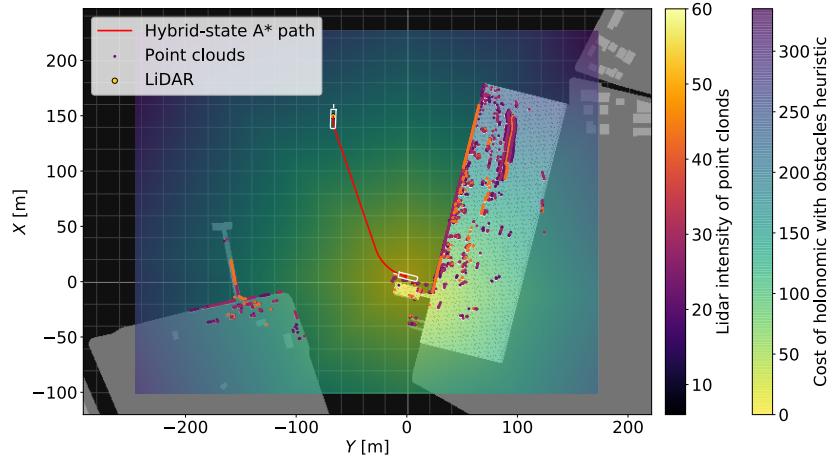


Figure 4.5 Path planning using Hybrid-state A* with point clouds measured by the LiDAR.

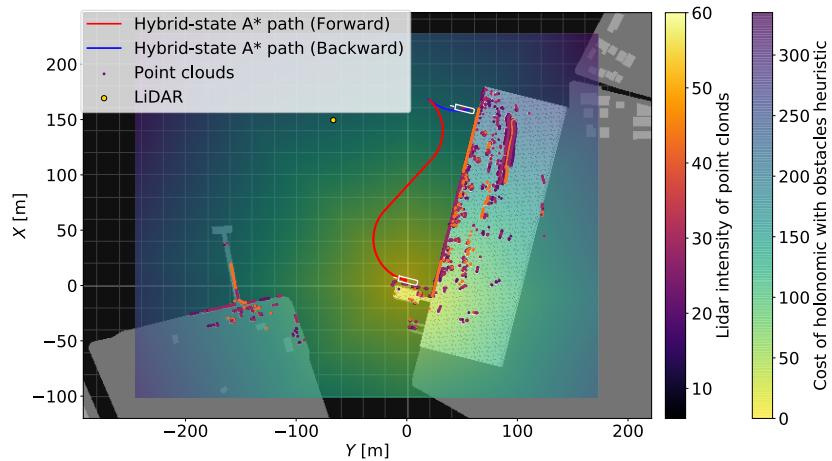


Figure 4.6 Path planning using Hybrid-state A* with point clouds measured by the LiDAR. (Collision avoidance).

4.3 海上における点群地図作成と自己位置推定

船体を自動制御する上で、船体位置の精度は重要であることは、これまでに述べたとおりである。筆者らは QZSS (Quasi-Zenith Satellite System, みちびき) に対応した受信機を用いて船体の位置を計測してきた。ここで使用したみちびき受信機は、センチメータ級の測位が可能な CLAS (センチメータ級測位補強情報) を利用することが可能であるが、一方で GNSS (Global Navigation Satellite System) に共通する問題として、受信状態によっては常に受信状態がセンチメータ級となる Fix 解になるとは限らず、場合によっては測位精度の悪い Float 解や単独測位になる場合がある点がある。Fig. 4.7は実験船の試験海域航行中のみちびき受信機の測位結果を表す GNSS クオリティの推移を示している。この図が示すように、航行中に常に Fix 解を得られるわけがないため、空の開けた海上でも測位精度が常に cm 級になるとは限らない。そのため、測位手段に冗長

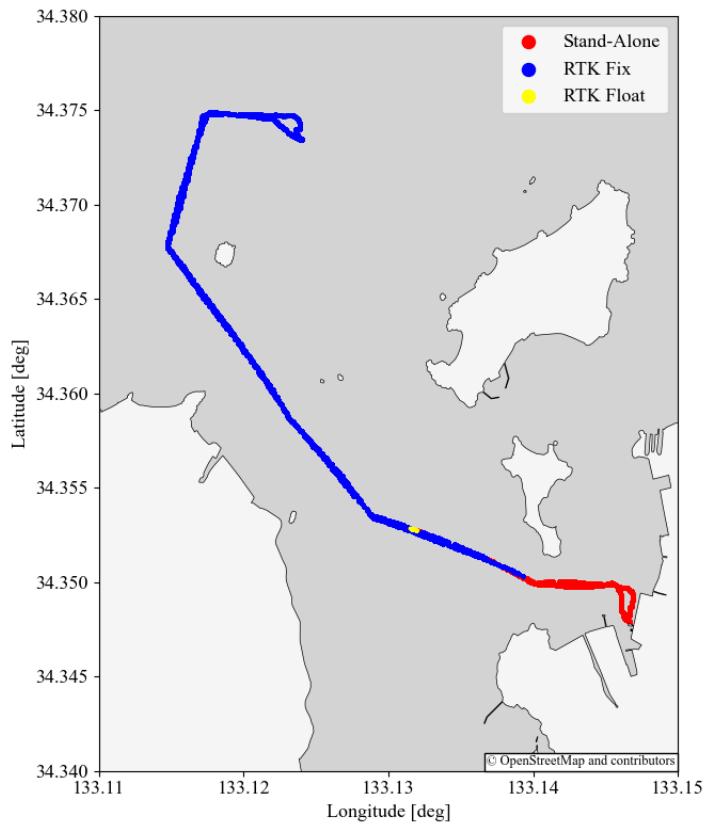


Figure 4.7 An example of measured GNSS quality data of the QZSS receiver around the test sea area.

性を持たせるために、別のセンサを組み合わせることは有効である。船首方位角を計測する GPS コンパスについても、内部で 2 つの GPS アンテナの側位結果から方位計算をしている関係上、同様の懸念から冗長化を検討する必要があるが、現状では GNSS を利用するもの以外で代替となる手段は少ない。

近年、SLAM（Simultaneous Localization and Mapping）という手法が、車の自動運転 [3, 4] や移動ロボットの分野で広く利用されている [5–8]。SLAM は、カメラやイメージセンサ、LiDAR（Light Detection and Ranging）などのセンサを使って周囲の環境情報を計測し、周辺環境の地図作成と自己位置推定を同時にを行う技術である。SLAM では、通常、移動体の水平方向の位置と向きを推定する。そのため、GPS 測位や GPS コンパスなどの GNSS を利用した手法の代替として利用されることがある [9]。本研究では、GNSS 以外の測位・方位角推定手段として、特に高精度な定位が要求される着桟時の利用を想定し、LiDAR を用いた自己位置推定・点群地図作成技術の可否を検討した。具体的には、LiDAR を用いた SLAM アルゴリズムを海上において適用し、桟橋周辺の海岸で点群地図作成、測位、方位角推定に関する性能検証を行った。

4.3.1 SLAM のアルゴリズム

繰り返しになるが、SLAM とは、移動体の自己位置の推定と環境地図の作成を同時にを行う技術のことである。SLAM の計算処理は、大きく 2 つに分けられる。一つは、LiDAR などのセンサで計

測されたデータを逐次処理するフロントエンドの部分と、もう一つはポーズグラフの最適化に重点を置くバックエンド部分である。本研究では、SLAM の性能を検証するために、2つのアルゴリズムを使用した。一つ目は、PCL (Point Cloud Library, [10]) を用いた NDT (Normal Distribution Transform[11]) スキャンマッチングのフロントエンド部分と、g²o (General Graph Optimization, [12]) によるグラフ最適化を用いたグラフベース SLAM のバックエンド部分からなる比較的単純なアルゴリズムである。NDT は、2つの点群をマッチングさせる手法の1つである。NDT では、計測した点群をボクセルに分割し、ボクセル内の点の分布を正規分布で近似して、新たに計測した点群データとのマッチングを行うことで、このマッチングでは、以下の評価関数を最小化する：

$$E(\mathbf{p}) = \sum_{i=0}^{N-1} \exp \frac{-(\mathbf{x}'_i - \mathbf{q}_i)^t \Sigma_i^{-1} (\mathbf{x}'_i - \mathbf{q}_i)}{2}, \quad (4.1)$$

ここで、 $\mathbf{p} = (t_x, t_y, \theta)^t$ は座標変換パラメータ、 \mathbf{x}_i は i 番目の入力スキャン点群データである。 \mathbf{x}'_i はパラメータ \mathbf{p} を用いて点群 \mathbf{x}_i を変換した後の点群である。 Σ_i と \mathbf{q}_i は、変換後の点群 \mathbf{x}_i の共分散行列と平均座標、 N は点数である。NDT の利点として、基準点群をボクセルに分割することでデータ量を削減できるため、全点間の距離を評価する ICP (Inerative Closest Point, [13]) に比べて一般に計算量が少ないことが挙げられる。バックエンド部のグラフ最適化は、主に地図の歪みを除去するループクロージャーを適用するために使用される。ループクロージャーは、センサノイズや逐次的なスキャンマッチングによって蓄積された誤差を修正する目的で使用される。ループクロージャーは、SLAM 実行中にセンサが環境内を計測した後、計測開始位置に戻ったときの移動した経路のループを検出し、ループの最初と最後の計測結果が同じ場所を指すようにマップ全体を修正する手法である。

この研究で使用されるもう一つのアルゴリズムは、地図の歪みを除去するために IMU を使用する LIO-SAM (tightly-coupled LiDAR Inertial Odometry via Smoothing and Mapping, [14]) である。LIO-SAM は、IMU の測定値をセンサの運動推定計算について、スキャンマッチングの前に情報を統合して点群の歪みを取り除き、LiDAR オドメトリの最適化のための初期推測を提供する。LIO-SAM は LeGO-LOAM[15] のコードをベースとしており、文献 [16] で提案されたスキャンマッチングアルゴリズムを採用している。LIO-SAM のフレームワークは、factor-graph[17] を用いて問題を定式化することで、IMU 計測値を事前統合している [18, 19]。ここで、具体的な計算手順を説明する。船体運動の状態ベクトル \mathbf{x} を次のように定義する：

$$\mathbf{x} = [\mathbf{R}^T, \mathbf{p}^T, \mathbf{v}^T, \mathbf{b}^T] \quad (4.2)$$

ここで、 $\mathbf{R} \in SO(3)$ は回転行列、 \mathbf{p} は位置ベクトル、 \mathbf{v} は速度、 \mathbf{b} は IMU 計測のバイアスである。世界座標系 \mathbf{W} から物体座標系 \mathbf{B} への変換行列 $\mathbf{T} \in SE(3)$ は、 $\mathbf{T} = [\mathbf{R}|\mathbf{p}]$ として定義する。IMU の角速度と加速度の計測値は次のように定義される：

$$\hat{\omega}_t = \omega_t + \mathbf{b}_t^\omega + \mathbf{n}_t^\omega \quad (4.3)$$

$$\hat{\mathbf{a}}_t = \mathbf{R}_t^{\mathbf{BW}} (\mathbf{a}_t - \mathbf{g}) + \mathbf{b}_t^\omega + \mathbf{n}_t^\omega, \quad (4.4)$$

ここで、 $\hat{\omega}_t$ and $\hat{\mathbf{a}}_t$ は時刻 t における座標系 \mathbf{B} における IMU の生の計測値である。 \mathbf{b}_t と \mathbf{n}_t は、時

刻 t における IMU のバイアスとホワイトノイズを表す。 \mathbf{R}_t^{BW} は W から B への回転行列を表す。 \mathbf{g} は W における定重直加速度ベクトルである。IMU の観測値は、2つのタイムステップ間の相対的な運動を用いて事前統合を行う [18] :

$$\Delta \mathbf{v}_{ij} = \mathbf{R}_i^T (\mathbf{v}_j - \mathbf{v}_i - \mathbf{g} \Delta t_{ij}) \quad (4.5)$$

$$\Delta \mathbf{p}_{ij} = \mathbf{R}_i^T (\mathbf{p}_j - \mathbf{p}_i - \mathbf{v}_i \Delta t_{ij} - \frac{1}{2} \mathbf{g} \Delta t_{ij}^2) \quad (4.6)$$

$$\Delta \mathbf{R}_{ij} = \mathbf{R}_i^T \mathbf{R}_j, \quad (4.7)$$

ここで、 $\mathbf{v}, \mathbf{p}, \mathbf{R}$ は船体の速度ベクトル、位置ベクトル、回転（回頭）運動の角速度ベクトルである。この IMU 事前統合の要素は、LIO-SAM の factor graph に組み込まれる。LiDAR 点群のキーフレームは変換演算に関連付けられ、ボクセルマップ \mathbf{M}_i に統合される。ボクセルマップは、LiDAR 点群のキーフレームを変換したエッジと平面の特徴量 ' \mathbf{F}_i^e , ' \mathbf{F}_i^p ' で構成される。LiDAR 点群から抽出されたエッジおよび平面特徴量間の距離は、次の式で計算できる。

$$d_{e_k} = \frac{\left| (\mathbf{p}_{i+1,k}^e - \mathbf{p}_{i,u}^e) \times (\mathbf{p}_{i+1,k}^e - \mathbf{p}_{i,v}^e) \right|}{\left| \mathbf{p}_{i,u}^e - \mathbf{p}_{i,v}^e \right|} \quad (4.8)$$

$$d_{p_k} = \frac{\left| (\mathbf{p}_{i+1,k}^p - \mathbf{p}_{i,u}^p) \cdot (\mathbf{p}_{i,u}^p - \mathbf{p}_{i,v}^p) \times (\mathbf{p}_{i,u}^p - \mathbf{p}_{i,w}^p) \right|}{\left| (\mathbf{p}_{i,u}^p - \mathbf{p}_{i,v}^p) \times (\mathbf{p}_{i,u}^p - \mathbf{p}_{i,w}^p) \right|}, \quad (4.9)$$

ここで k, u, v, w は特徴量のインデックスであり、 $\mathbf{p}_{i+1,k}^e$ は点 $\mathbf{p}_{i,u}^e$ と点 $\mathbf{p}_{i,v}^e$ で構成されるエッジに対応する W 上の平面特徴量 ' \mathbf{F}_{i+1}^e ' におけるエッジ特徴量である。' \mathbf{F}_{i+1}^p ' における平面特徴量 $\mathbf{p}_{i+1,k}^p$ について、 $\mathbf{p}_{i,u}^p, \mathbf{p}_{i,v}^p, \mathbf{p}_{i,w}^p$ は対応する平面 \mathbf{M}_i^p に含まれる点である。ここで、最適な変換 \mathbf{T}_{i+1} は次式を最小化することで得られる:

$$\min_{\mathbf{T}_{i+1}} \left\{ \sum_{\mathbf{p}_{i+1,k}^e \in 'F_{i+1}^e} d_{e_k} + \sum_{\mathbf{p}_{i+1,k}^p \in 'F_{i+1}^p} d_{p_k} \right\}. \quad (4.10)$$

変換 \mathbf{T}_i を用いて、船体運動の状態ベクトルを得られる：

$$\mathbf{x}_{i+1} = \Delta \mathbf{T}_{i,i+1} \mathbf{x}_i, \quad (4.11)$$

ここで、 $\Delta \mathbf{T}_{i,i+1} = \mathbf{T}_i^T \mathbf{T}_{i+1}$ である。以上がアルゴリズムの概要である。詳細については、原論文 [14] を参照されたい。

4.3.2 少ない水平な平面特徴量に対する SLAM

海上での SLAM は、海岸付近の水深が浅いことが多く船舶が近づけないため海岸からの計測距離が長くなる傾向にあり、計測できる LiDAR 点群の数が少なくなるため難しいとされている。さらに、海上での SLAM は、地表面や海面の点群をほとんど取得できないため、水平方向の平面形

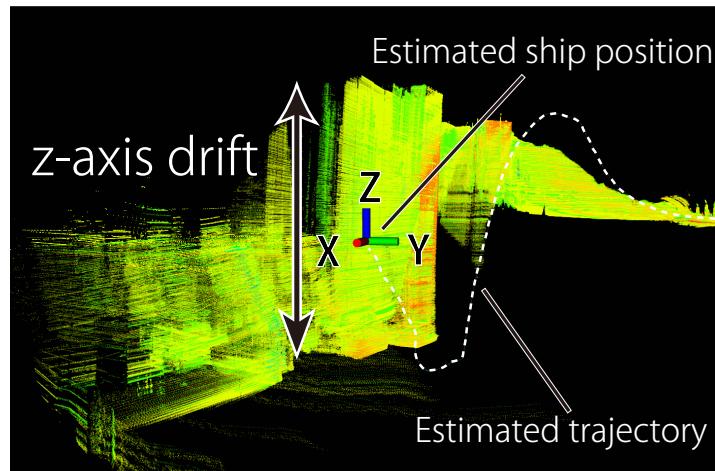


Figure 4.8 Z-axis drift of scan matching on the sea

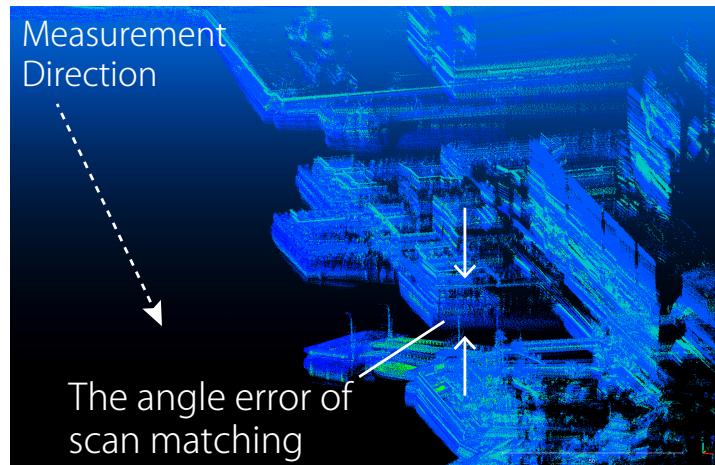


Figure 4.9 The error in angular transformation of scan matching in the roll direction of the ship

状が十分に得られないという課題もある。これは単に取得できる点群の数が少ないという問題だけに収まらない。十分な点群が取得できたとしても、水平位置や yaw 角の座標変換はできるものの、水平方向の平面形状がないため、垂直方向やロール・ピッチ方向のマッチングに誤差が生じやすい。ここでは、海上での SLAM におけるスキャンマッチングの失敗の典型的な例として、z 軸方向のドリフトとロールに関する誤差について説明する。まず Fig. 4.8では、垂直 Z 軸方向の点群マッチングの誤差が生じる様子を示している。この点群マッチングの失敗により、推定された船の軌跡も垂直方向に現実にはあり得ない航跡を描いており、航跡がドリフトしている。Fig.4.9では、計測方向に対してロール方向にマッチング誤差が発生し、点群マップが二重に重なって見えている。この誤差はピッチ方向でも同様に発生することがある。このようなスキャンマッチング計算の誤差の原因を Fig. 4.10に示す。図中 (b), (c) の例では、水平方向の平面形状が得られず、船のヒープ方向の変位とロール角が決定できない。本研究で使用した LIO-SAM は、LiDAR のオドメトリを最適化するために IMU の観測値を利用し、スキャンした点群の歪みを取り除く

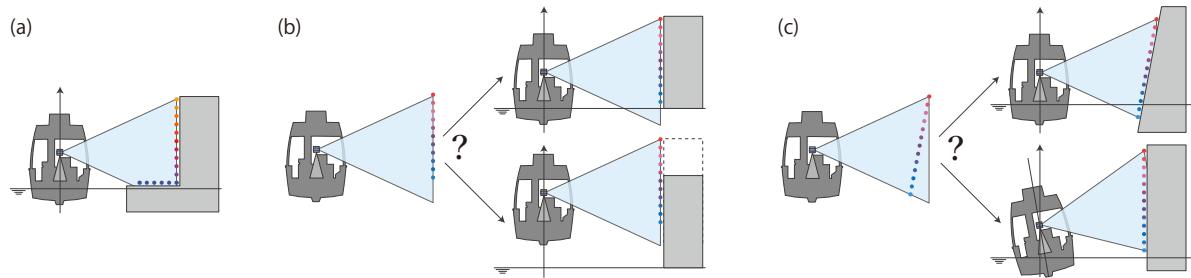


Figure 4.10 Examples of SLAM failures at Sea. (a) is an example where horizontal planar features are obtained with LiDAR scan, in which case the position and angle of the ship (LiDAR) in the vertical plane can be determined. (b) and (c) are examples where horizontal planar features aren't obtained. In (b), the heave displacement of the ship (LiDAR) cannot be determined because the true height of the object is not known from the acquired point clouds. In (c), it is not possible to determine whether the object being measured is tilted or the ship (LiDAR) is tilted because it is not known how much the obtained point cloud is tilted in the space-fixed coordinates.

ことでスキャンマッチングの初期推測値を提供する。このため、IMU の観測値は最適化の初期値に影響するだけで、最適化の結果 Fig. 4.10の例のようなマッチングの失敗が発生する問題を根本的に解決することはできない。そこで、海上の水平な平面特徴量の欠落によるスキャンマッチングの失敗を抑制するために、SLAM 計算において次の座標変換制約を導入した：1) z 軸座標変換 T_z に対して、 $T_z < 1.0 [m]$, 2) ロール角とピッチ角の座標変換 T_{roll} T_{pitch} について、 T_{roll}, T_{pitch} は $T_{roll}, T_{pitch} < 10 [deg]$ とした。制約条件のパラメータは、実際に測定したデータに基づいて決定した。これらの制約を上記より小さい値でも試したが、結果の精度に影響はなかった。さらに、点群のスキャンマッチング時にノイズとなる自船の点群や船の航跡を含まないように、LiDAR の周囲 30 [m] の点群を除外した。

4.3.3 実験に使用したセンサ、ソフトウェア、計測機器



Figure 4.11 Sensor configuration of LiDAR and IMU

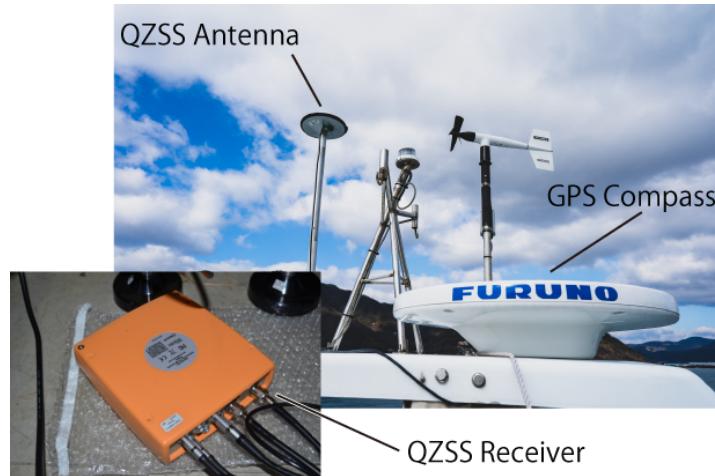


Figure 4.12 Sensor configuration of GNSS (GPS and QZSS)

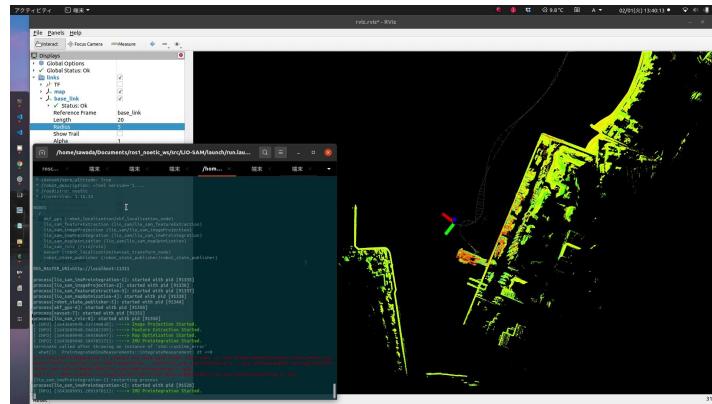


Figure 4.13 Screenshot of LiDAR SLAM (LIO-SAM).

本研究でSLAMに使用する計測データ収集には、実験船「神峰」を使用した。本研究で使用したLiDARセンサはVelodyne社のVLP-32C LiDARで、三脚を使用して甲板に設置した。LiDARの値を補正するために使用したIMUは、9軸センサ（3軸加速度+3軸ジャイロ+3軸磁力計）を持つWitMotion社のセンサである。9軸センサの出力周波数は50Hzに設定した。9軸センサによるLiDAR本体の動きを計測するため、IMUの地磁気センサへの影響を考慮し、金属製のマウントではなく、FDM（Fused Deposition Modeling）方式の3Dプリンターを用いてPLA樹脂で作成したマウントを使用した。Fig. 4.11は、実験におけるセンサの取り付けの様子である。LiDAR SLAMの結果を評価するために、Fig. 4.12に示すような船舶用GPSコンパスとQZSS受信機を使用した。QZSS受信機は、GPSコンパス付近と実験船の船首付近の設置した2つのアンテナに接続し、船の位置と方位角を計測できるようにした。LiDARとIMUのデータは、SLAMに使用するため、ROS（Robot Operating System）のデータログ形式であるrosbagとして記録した。GPSとQZSSの計測値は、船上のPLCシステムを使ってLiDARの計測に合わせて記録した。

本論文のSLAMアルゴリズムは、ロボット開発に広く利用されているROS上で実行した。NDTベースのSLAMは、ROS 2 Galactic上で実行した。LIO-SAMは、ROS 1 Noeticで実行した。い

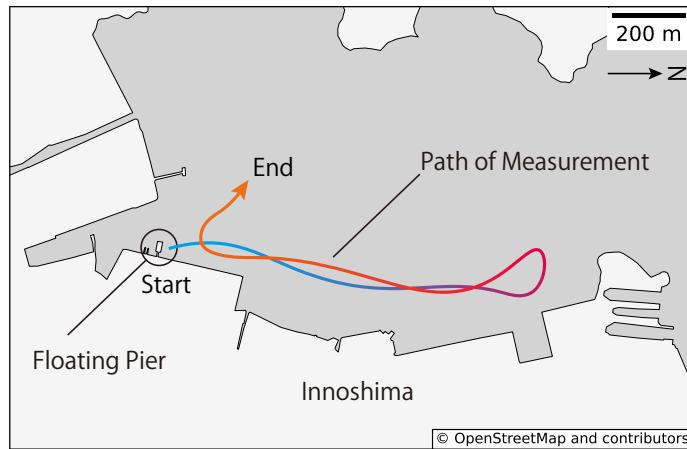


Figure 4.14 Location and trajectory of measurement.

ずれも Ubuntu 20.04 LTS を搭載し、Core i7-9750H プロセッサと 16GB RAM を装備したノート PC 上で実行した。Fig. 4.13に、ノートパソコン上で動作する LIO-SAM の画面を示す。画面にはスキャンマッチングとグラフ最適化の計算ログが出力され、可視化ソフトウェア Rviz には現在作成中のマップが表示される。今回は、作成した地図と自己位置推定結果を出力するコマンドを手動で送り、解析を行った。作成したマップは、Fig. 4.14に示すように、浮桟橋からドックのある浮桟橋北側の海岸を往復するルートで作成した。測定中の船の平均地上速度は 4.2 [kt] であった。

4.3.4 結果

4.3.4.1 点群地図の作成

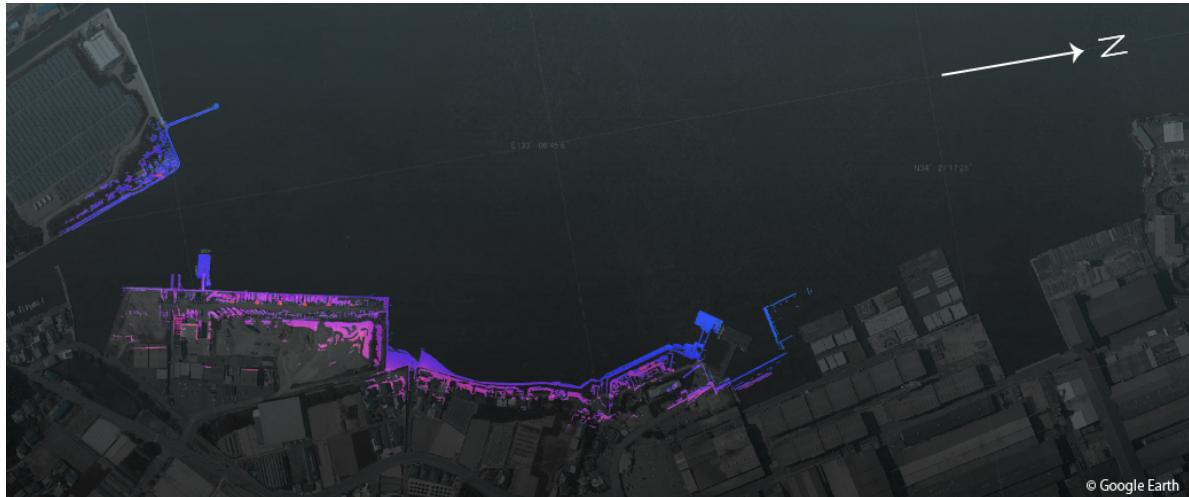


Figure 4.15 Point cloud map generated using the NDT based algorithm without an IMU. The background aerial photograph is referred to Google Earth.

実験船を用いて計測したデータに対して、NDT ベースのアルゴリズムと LIO-SAM による

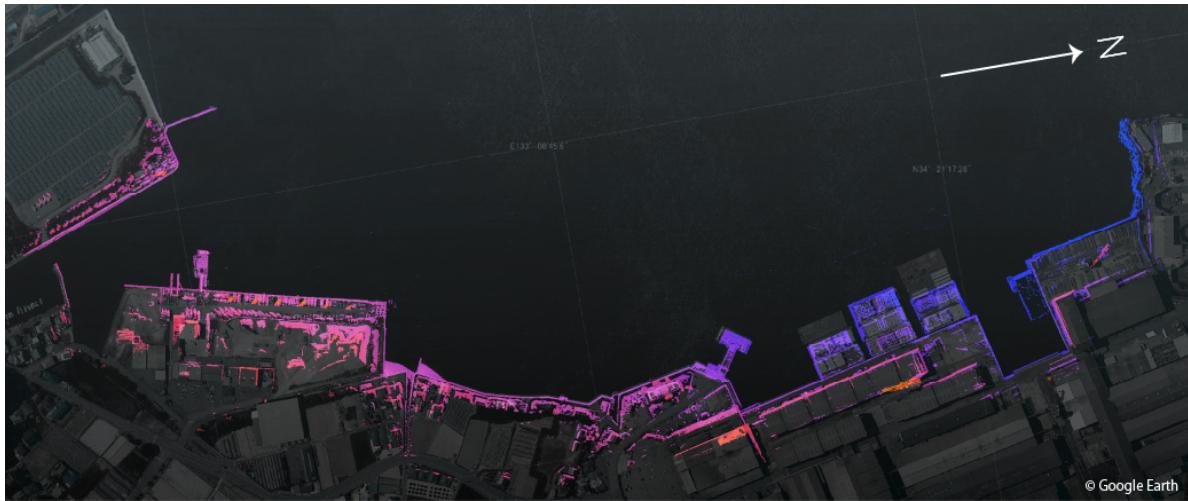


Figure 4.16 Point cloud map generated using the LIO-SAM algorithm. The background aerial photograph is referred to Google Earth.

SLAMを実施した。本研究で作成した点群マップを、Fig. 4.15とFig. 4.16に示す。比較のために点群マップをGoogle Earthの航空写真に重ね合わせた。図中のSLAMを行った2つのケースは、IMUなしとIMUありの異なる計測データであり、計測経路は図中に示したものとほぼ同じである。IMUを用いないNDTベースのアルゴリズムでは、計測開始地点の因島マリーナがある浮桟橋から離れると、地図の歪みが大きくなっている。この例では、地図の歪みが大きいために地図作成が途中で失敗したため、地図の歪みを除去するループクロージャーも行うことができなかった。一方でLIO-SAMを用いたSLAMの結果では、SLAM計算中に何度もループクロージャーが働き、マップの歪みが適切に除去された。結果として、スキャンマッチングにおける座標変換の拘束条件を導入したLIO-SAMの方がより正確な地図が作成された。

標準的な船の場合は、ECDIS(電子海図表示情報システム)などのENC(電子航海図)やレーダー(電波探知機・測距儀)などが使用される。自動運航船の場合も、同様に海図を使って船の位置や進路をモニター画面で確認することが想定される。自動着桟制御については、OpenStreetMapで作成した地図データを3D地図に変換して、自動着桟制御用のGUIを有するAutoBerthを開発している。ここでは、この3D地図にLIO-SAMで作成した点群地図を重ね合わせ、両者を比較する。Fig. 4.17にAutoBerthの画面を示し、その下に浮桟橋周辺の拡大画像を示す。画像からもわかるように、点群と3Dマップは概ね一致している。しかし、浮き桟橋周辺の点群を見ると、地図と位置がずれていることがわかる。また、浮き桟橋の左側に突き出ている船を陸にあげるための桟の位置も異なっている。これらの地図上の形状は、OpenStreetMapだけでなく、Googleマップや国土地理院が作成した地図など、他の地図データでも同様である。後者の2つの地図データも、そもそも浮桟橋は地図上に存在しない。精度の高い電子海図はオープンデータではないため、例えば障害物回避や進路計画などの自律航行船のタスクの研究開発に利用する点で問題がある。一方でオープンな地図データは、経路計算や監視用のGUIには使いやすいが、精度に問題がある。LiDARで高精度な地図が作成できれば、前節のような港湾形状を考慮した進路計画などへの応用が期待される。

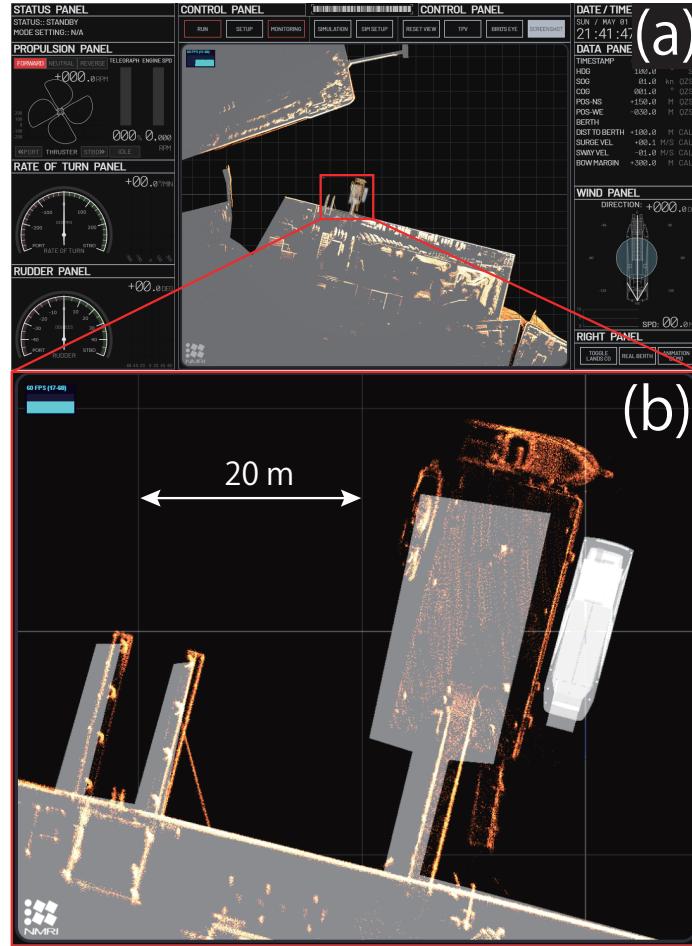


Figure 4.17 (a) The point cloud map created using LIO-SAM superimposed on the 3D map data in the automatic berthing control GUI. (b) Enlarged image of the area around the floating pier. The original map data of 3D map is referred to OpenStreetMap.

4.3.4.2 自己位置推定

本節では、LiDAR SLAM (LIO-SAM) の自己位置推定と GPS, QZSS の測位性能を比較する。測定時の軌跡を Fig. 4.18に示す。測定中の QZSS の測位情報は、常に RTK (Real-Time Kinematic) 測位の Fix 解の状態であった。図中の位置は、LiDAR と GPS/QZSS アンテナの船上位置から算出した船のミッドシップ位置を表している。ただし、以下の図における船の方位角は、SLAM では自己位置推定結果、GPS では GPS コンパス、QZSS では QZSS コンパスからそれぞれ推定もしくは取得している。したがって、これらのセンサからの方位角データの精度が、ミッドシップ位置推定の誤差に影響することに留意する必要がある。緯度方向に 600 [m] 以下の範囲内では、3 つの方法で得られた航跡はよく一致している。しかし、図中の桟橋付近の拡大図を見ると、桟橋付近の軌跡は、3 つの方法で得られた軌跡とよく一致している。Fig. 4.18では、GPS の軌道が他の 2 つの軌道から大きくずれている部分があることがわかる。一方、SLAM と QZSS の軌道はほ

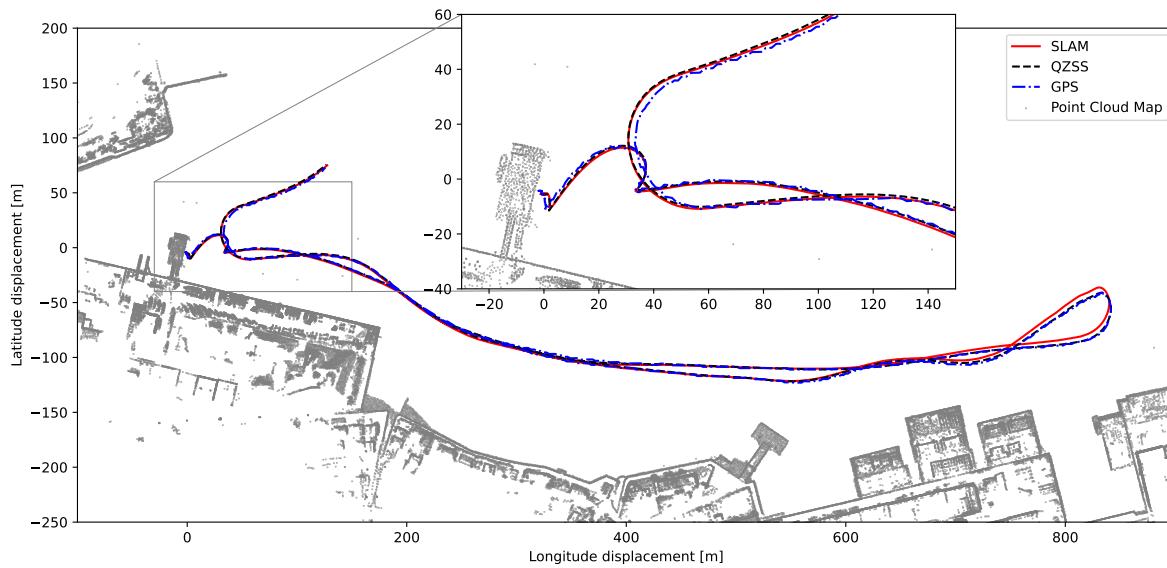


Figure 4.18 Comparison of trajectories acquired by SLAM, QZSS and GPS. The point cloud map in this figure was downsampled for plotting.

とんど一致している。また、GPSによる軌道は全体的に滑らかでないのに対し、SLAMとQZSSの軌道は滑らかである。

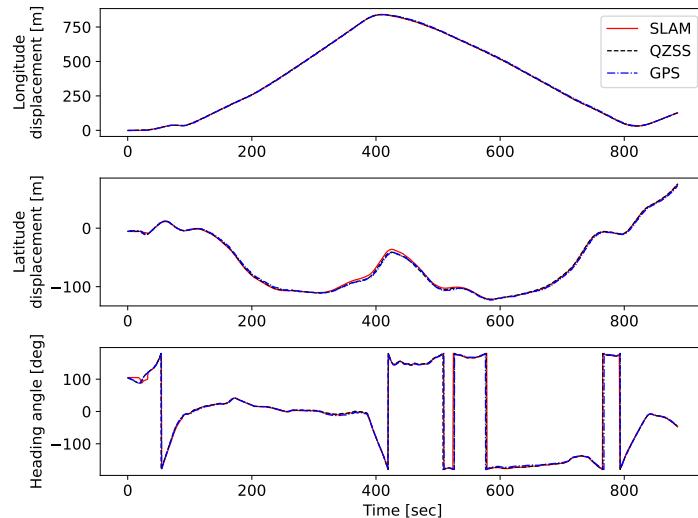


Figure 4.19 Time series of the displacement estimated using SLAM (LIO-SAM) and GPS/QZSS positioning and the heading angle estimated using SLAM(LIO-SAM) and the GPS/QZSS compass.

次に、SLAM、GPS、QZSSで推定・計測した緯度・経度変位と方位角の時系列をFig. 4.19に示す。方位角はグラフを見やすくするために、北を0度、東を正、西を負として、 ± 180 度の範囲で示した。Fig. 4.19では、3つの方法で得られた値は、ほとんどの区間で概ね一致していることがわかる。しかし、Fig. 4.19を見ると、SLAMはGPS/QZSSよりも誤差が大きく、特に測定開始

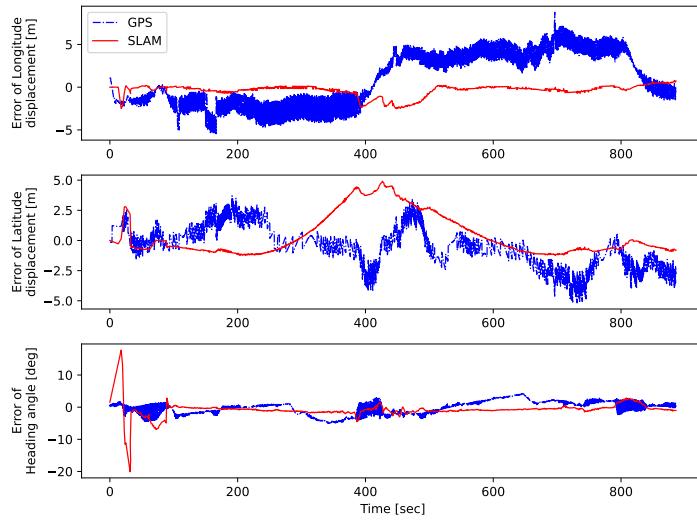


Figure 4.20 Error of the displacement using SLAM (LIO-SAM) and GPS positioning versus QZSS positioning, and error of the heading angle estimated using SLAM (LIO-SAM) and the GPS compass versus the QZSS compass.

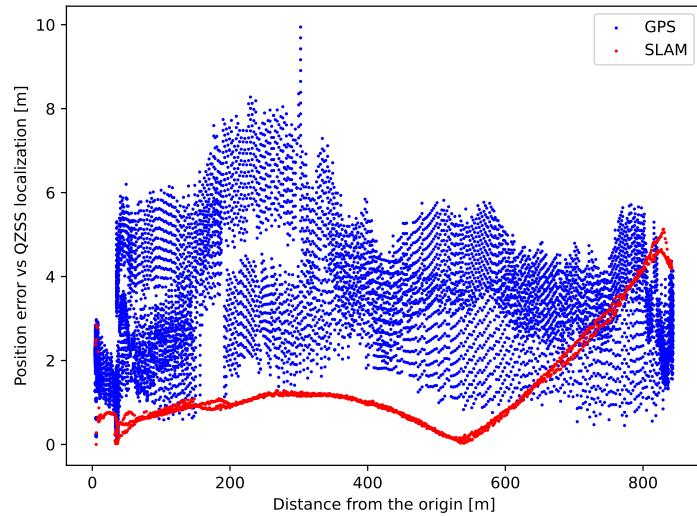


Figure 4.21 Position error of SLAM and GPS vs QZSS.

時の方位角の誤差が大きいことがわかる。この方位角の推定誤差は、SLAM の計算開始直後のマッピングが不完全であったためと推測される。

時系列を単純に比較しただけでは、それぞれの推定値の誤差がわかりにくいため、RTK Fix 解である QZSS の値を基準としたときの差分を Fig. 4.20 に示す。この図では、GPS と SLAM の QZSS に対する差が明確に示されている。図中の SLAM による変位推定値の差は、緯度方向の差よりも経度方向の差の方が大きくなっている。GPS の結果は SLAM や準天頂衛星に比べてノイズが多く、測位結果にはらつきがある。緯度方向のズレを見ると、GPS の誤差が移動方向と逆に発生していることも読み取れ、これは GPS の測位結果が遅れていることを示している。船首方位角については、SLAM の方が船用の GPS コンパスが output する値と比較して安定した推定値を出力で

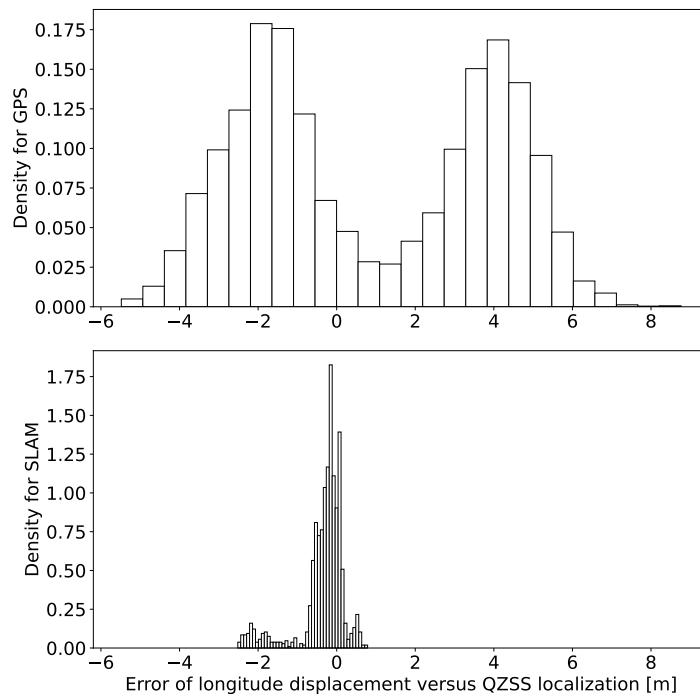


Figure 4.22 Probability density of longitude displacement error of SLAM and GPS versus QZSS.

きている。特に、測定中の3度ある旋回の区間では、SLAM、GPSともに方位角データにノイズが発生しているが、SLAMは船舶用GPSコンパスと同程度の精度で推定している。ただし、測定開始時の偏差は最大20[deg]程度であり、SLAMの使用では、計測開始後に船首方位角の推定値が安定するまで待つことが推奨される。

Fig. 4.21は座標原点からの距離を横軸に、GPSと準天頂衛星測位データに基づくSLAM結果の誤差を比較したものである。SLAMの位置誤差が、計測の行きと帰りで原点からの距離に対して一定であることは、SLAMによる自己位置推定そのものが安定していることを示している。原点から600[m]以内の区間では、SLAMの位置とQZSSの位置の誤差は、ほとんど1.3[m]の範囲内に収まっている。また、GPSについては、Fig. 4.18の軌跡を重ね合わせると、全体的にGPSとQZSSの軌跡がよく一致しているように見えるが、実は各時点で取得した位置データには大きな誤差があり、誤差の平均と分散は全区間にわたり3.0[m], 1.5[m]となった。GPSの位置情報と比較すると、SLAMによる自己位置推定は位置データのばらつきが小さくなる傾向にある。

最後に、各データセットの誤差の分布について見る。例によって、GPSとSLAMの結果を準天頂衛星のデータに関して比較している。Fig. 4.22, 4.23, 4.24, 4.25に、SLAMとGPSの経度変位、緯度変位、位置、方位角の誤差の密度分布をそれぞれQZSSと比較した結果を示す。まず、緯度変位については、GPSの誤差分布は明らかに2つのピークを示し、推定の遅れが原因であることが考えられる。ここで、GPSと比較して、SLAMの緯度変位の分散は非常に小さい。経度変位の分布は、この場合、測定の経路のおかげでピークが1つになっている。一方、SLAMの分布の誤差は、地図の歪みによる誤差のため、プラス側に寄っている。位置の面で2つの誤差を比較すると、SLAMは経度変位の誤差の影響を受けるが、その分布はGPSよりも原点に近い位置にあ

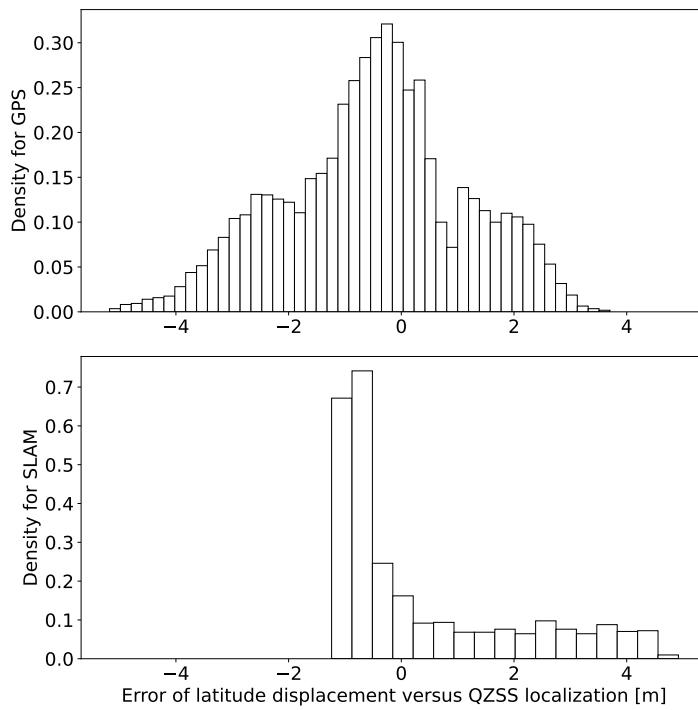


Figure 4.23 Probability density of latitude displacement error of SLAM and GPS versus QZSS.

る。船首方位角については、GPS の誤差分布は SLAM の誤差分布よりも広がっており、SLAM の方がより正確であることがわかる。また、SLAM の分布はわずかに負のバイアスを示している。Table 4.1 は、経度緯度変位、位置、方位角について、GPS と SLAM と QZSS の誤差の統計量をまとめたものである。太字の値は、GPS と SLAM の結果のうち、より良い方を表している。より多くの項目で、SLAM の推定値が GPS よりも精度が高い。実際の着桟制御では、桟橋付近の位置データの精度が重要である。そこで、桟橋から 600[m] 以内のデータについても SLAM の結果を分析した。桟橋から 600 [m] 以内の平均位置誤差は 0.79 [m]、標準偏差は 0.36 [m] でした。緯度変位については、桟橋から 600 [m] 以内の誤差は平均-0.63 [m]、標準偏差は 0.51 [m] であった。これらの結果は、LiDAR SLAM が、冒頭で述べた自動着桟制御に必要な精度を満たすことができるることを示している。

4.4 岸壁検知システムの開発

本章の最後に、岸壁検知システムの開発について説明する。

まず問題設定として、PLC を使った組み込みシステムとして LiDAR 点群を用いた岸壁検知を行うことを目的とする。ここで、前節で実施した SLAM の課題として、現状 SLAM の計算が複雑かつ計算量が膨大なため、GPU を搭載した高性能な PC 上でしか動作できない問題がある。これは、AutoBerth PLC の項でも挙げた問題と共にし、実用的な形で実装を行う手段として、ここでも PLC への実装を考える。一方で、PLC の計算能力では、SLAM のような計算を行うことはできない。そこで、取得した点群から特徴点を抽出して岸壁を検出し、船体と岸壁との間の距離

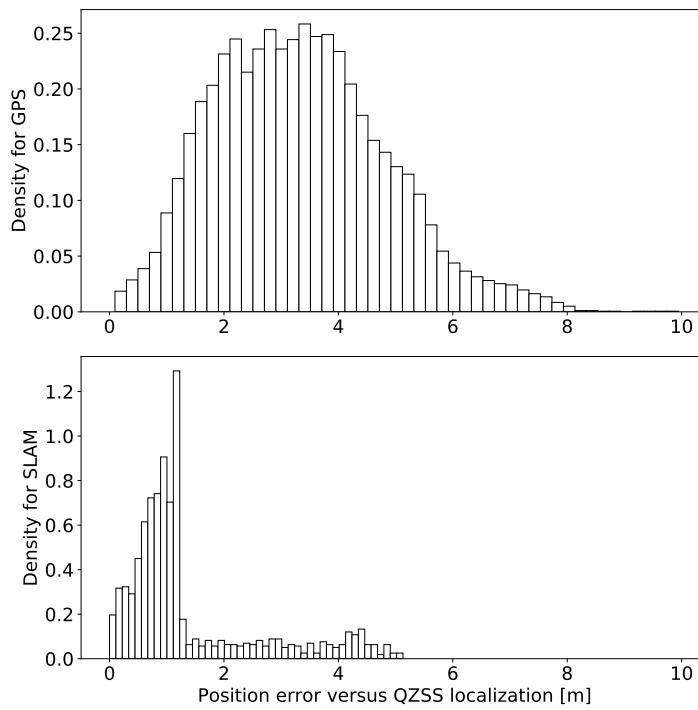


Figure 4.24 Probability density of position error of SLAM and GPS versus QZSS.

Table 4.1 Statistics of error of GPS and SLAM compared with QZSS

subject	average μ	standard deviation σ
Displacement		
Longitude (GPS) [m]	0.90	3.1
Longitude (SLAM) [m]	-0.34	0.59
Latitude		
Latitude (GPS) [m]	-0.50	1.6
Latitude (SLAM) [m]	0.32	1.67
Position		
Position (GPS) [m]	3.3	1.5
Position (SLAM) [m]	1.4	1.2
Heading angle		
Heading angle (GPS) [deg]	-0.19	2.0
Heading angle (SLAM) [deg]	-0.90	1.5

を計測するシステムとして実装する。

実際に、ケミカルタンカーに設置したLiDARを用いて公共岸壁を計測したものがFig. 4.26である。岸壁の検知とは、点群の中に直線的に並ぶ岸壁の点群に対して、その直線の位置や傾き等のパラメータを推定することとする。

計算量を抑えつつ、ノイズにも耐えるロバストなシステムにするために、今回はRANSAC(Random Sample Consensus)アルゴリズムを採用した。今回実装したRANSACは次のように直線検知を行う：

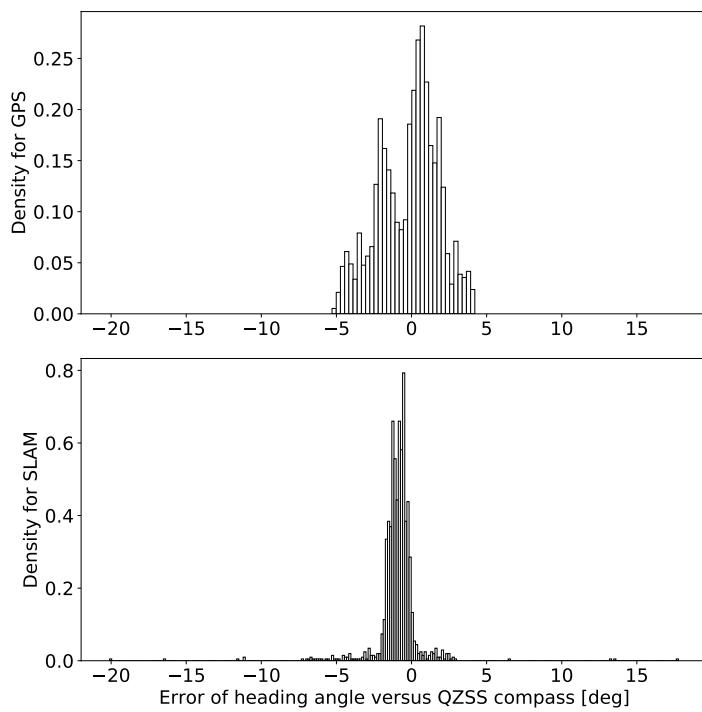


Figure 4.25 Probability density of error of heading angle estimated using LIO-SAM and the GPS compass versus the QZSS compass.

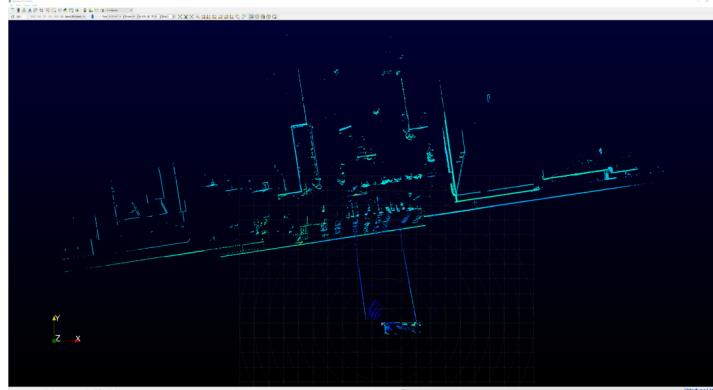


Figure 4.26 Point clouds of the berth measured on the chemical tanker.

1. ある時刻に計測された点群データから、直線検知に用いるデータをランダムにサンプリングする。
2. サンプリングしたデータ点に対して最小二乗法により、直線を推定する。
3. すべてのデータに対して、予めした距離より近いデータ点に対して推定した直線との誤差を計算する。
4. 前の試行で推定した直線より誤差が小さい場合は、直線モデルを更新する。
5. 以上を繰り返し、規定の回数に達したら終了する。

このようにして、より誤差の小さい直線を推定することで、岸壁の点群に直線をフィッティング

させて岸壁を検知する。PLC では、一回の計算ループにおいて許容される計算時間が決まっているため、計算時間に達した時点でも RANSAC の計算を打ち切る。その場合、船上の LiDAR の設置の向きから、推定された岸壁が規定値よりも大きな傾きで推定された場合は、異常値として推定された直線を棄却する。また、PLC における RANSAC の計算量を減らすために、船上から水平方向にある岸壁に当たらない方向に飛んだレーザーの点群を計算に含まないように、計算に使う LiDAR のレーザー仰角を 3 つに限定して計算を行う。そのため、実質的には 3 ラインの LiDAR であれば岸壁の検知を実施できる。

実際に実装した PLC システムを Fig. 4.27 に示す。また、計測に用いた LiDAR の設置の様子を Fig. 4.28 に示す。着桟操船中の岸壁検知の様子を、Fig. 4.29 に示す。画面には、選択した LiDAR の仰角の情報と計測された点群が表示され、推定された岸壁の直線が重畠されている。また、推定された直線と船首尾の距離をリアルタイムに表示することができる。本研究では、PLC による岸壁検知のプロトタイプを作成した。より、実用的なシステムにするために、自動着桟のための AutoBerth PLC への接続や、船橋内での情報提供などといったことが将来の課題である。



Figure 4.27 PLC system of berth line detection.

4.5 結言

本章では、センサ系の冗長化を通じた自動着桟システムの高度化のために、LiDAR を適用した 3 つの研究成果を示した。1 つ目は、岸壁等の障害物を考慮した経路計画である。この研究では、1 フレーム分の点群データを用いて、切り返し操船を含む高度な経路計画を行うことができた。今後は、次の成果である LiDAR SLAM による点群地図を用いたより実用的なシステムの開発を行うことが課題である。次の研究成果は、海上における LiDAR SLAM に関するものである。まず、点群地図作成の過程で計測した点群をつなぎ合わせるスキャンマッチングの計算において、海上での適用における問題点を示した。そこで、スキャンマッチングにおける拘束条件の導入を提案することで、実際に因島マリーナ周辺の海岸に沿った点群の計測データに対して、拘束条件を導入した LIO-SAM や広く利用される NDT と比べてより広範囲の点群地図を高精度に作成することができた。また、QZSS と GPS の計測結果と LiDAR SLAM の自己位置推定の結果を比較し、



Figure 4.28 LiDAR (on left) for berth line detection.

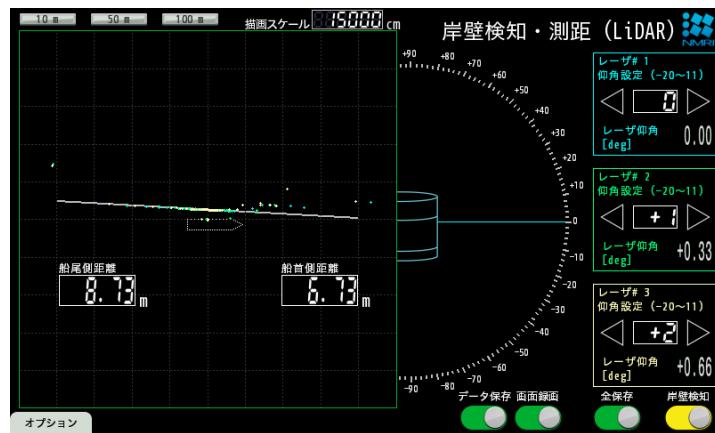


Figure 4.29 Screenshot of the line detection of the berth using the proposed system.

計測開始位置から 600 [m] 以内の範囲において、QZSS と GPS に匹敵する高精度な推定を行うことができた。最後の研究成果は、LiDAR と PLC により構成される岸壁検知システムの開発である。PLC は、リアルタイム性能を実現するにあたり、一般的な PC と比較して計算資源が限定されているが、そのような機器でリアルタイムにかつ安定的に岸壁を検知するべく、RANSAC アルゴリズムを用いた岸壁検知システムを実装した。ケミカルタンカーの着桟操船中に、本システムを用いて岸壁検知を行い、安全な着桟制御に必要な岸壁と船首尾間の距離を計測できることを示した。これらの成果を今後自動着桟システムへと統合するとともに、これに伴うデータ監視などの安全機構やデータフュージョンの機能の実現は今後の課題である。次の章では、提案する自動着桟システムの制御アルゴリズムを数値シミュレーションおよび実船実験により検証する。

第4章の参考文献

- [1] D. Dolgov, S. Thrun, M. Montemerlo, J. Diebel, Practical search techniques in path planning for autonomous driving. in *Proceedings of the First International Symposium on Search Techniques in Artificial Intelligence and Robotics (STAIR-08)* (AAAI, Chicago, USA, 2008)
- [2] J. Reeds, L. Shepp, Optimal paths for a car that goes both forwards and backwards, *Pacific Journal of Mathematics* **145**(2), 367 (1990)
- [3] S. Kato, S. Tokunaga, Y. Maruyama, S. Maeda, M. Hirabayashi, Y. Kitsukawa, A. Monrroy, T. Ando, Y. Fujii, T. Azumi, Autoware on board: Enabling autonomous vehicles with embedded systems. in *2018 ACM/IEEE 9th International Conference on Cyber-Physical Systems (ICCPs)* (2018), pp. 287–296
- [4] M. Elhousni, X. Huang, A survey on 3D lidar localization for autonomous vehicles. in *2020 IEEE Intelligent Vehicles Symposium (IV)* (2020), pp. 1879–1884
- [5] M. Filipenko, I. Afanasyev, Comparison of various slam systems for mobile robot in an indoor environment. in *2018 International Conference on Intelligent Systems (IS)* (2018), pp. 400–407
- [6] T. Dang, M. Tranzatto, S. Khattak, F. Mascarich, K. Alexis, M. Hutter, Graph-based subterranean exploration path planning using aerial and legged robots, *Journal of Field Robotics* **37**(8), 1363 (2020), wiley Online Library
- [7] M. Kulkarni, M. Dharmadhikari, M. Tranzatto, S. Zimmermann, V. Reijgwart, P.D. Petris, H. Nguyen, N. Khedekar, C. Papachristos, L. Ott, R. Siegwart, M. Hutter, K. Alexis, Autonomous teamed exploration of subterranean environments using legged and aerial robots. in *2022 IEEE International Conference on Robotics and Automation (ICRA)* (IEEE, Philadelphia (PA), USA, 2022)
- [8] T. Miki, L. Wellhausen, R. Grandia, F. Jenelten, T. Homberger, M. Hutter. Elevation mapping for locomotion and navigation using GPU (2022). DOI 10.48550/ARXIV.2204.12876. URL <https://arxiv.org/abs/2204.12876>
- [9] G. Grisetti, R. Kümmerle, C. Stachniss, W. Burgard, A tutorial on graph-based slam, *IEEE Intelligent Transportation Systems Magazine* **2**(4), 31 (2010)
- [10] R.B. Rusu, S. Cousins, 3D is here: Point Cloud Library (PCL). in *IEEE International Conference on Robotics and Automation (ICRA)* (Shanghai, China, 2011)
- [11] P. Biber, W. Strasser, The normal distributions transform: a new approach to laser scan matching. in *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003)*, vol. 3 (2003), vol. 3, pp. 2743–2748 vol.3
- [12] R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, W. Burgard, g²o: A general framework for graph optimization. in *2011 IEEE International Conference on Robotics and Automation* (2011), pp. 3607–3613
- [13] P. Besl, N.D. McKay, A method for registration of 3-d shapes, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **14**(2), 239 (1992)
- [14] T. Shan, B. Englot, D. Meyers, W. Wang, C. Ratti, D. Rus, LIO-SAM: Tightly-coupled lidar

- inertial odometry via smoothing and mapping. in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (2020), pp. 5135–5142
- [15] T. Shan, B. Englot, LeGO-LOAM: Lightweight and ground-optimized lidar odometry and mapping on variable terrain. in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (IEEE, 2018), pp. 4758–4765
 - [16] J. Zhang, S. Singh, Low-drift and real-time lidar odometry and mapping, *Autonomous Robots* **41**(2), 401 (2017)
 - [17] F. Dellaert, M. Kaess, *Factor Graphs for Robot Perception* (Now Publishers Inc., 2017)
 - [18] C. Forster, L. Carlone, F. Dellaert, D. Scaramuzza, Imu preintegration on manifold for efficient visual-inertial maximum-a-posteriori estimation. in *Robotics: Science and Systems (RSS)* (2015)
 - [19] F. Dellaert, Factor graphs and gtsam: A hands-on introduction (2012)

第 5 章

自動着桟制御

5.1 緒言

本章では、本研究で提案する自動着桟制御アルゴリズムの検証を行う。自動着桟アルゴリズムとして、PPA 制御アルゴリズムおよび FFPPA 制御アルゴリズムを対象として、数値シミュレーションおよび実船実験により検証を行う。また、安全なアプローチ角度を目的とした、経路追従時の目標方位角の補正の導入と経路追従制御の初期に発生する経路偏差を軽減するための助走区間の導入についても、検証結果を示す。さらに、今回実装した AutoBerth (PC) と AutoBerth PLC の比較として、PC と PLC での制御の比較も行う。なお、本章で行う数値シミュレーションに関して、対象船の操縦運動数学モデルと風況の再現方法については、本論の Appendix として末尾に記載している。

5.2 数値実験および実船実験の概要

5.2.1 試験海域



Figure 5.1 Test field of automatic berthing experiments

提案する自動着桟制御手法を検証するための実船試験は、因島マリーナ（広島市尾道因島）周

辺の海域において実施した (Fig. 5.1). 本海域は、マリーナの浮桟橋周辺に浅瀬が存在し、着桟経路が制限される。今回は、浮き桟橋の北側に右舷付けで着桟する。数値シミュレーションもこの海域に準じて計算を実施した。

5.2.2 着桟制御の目標設定

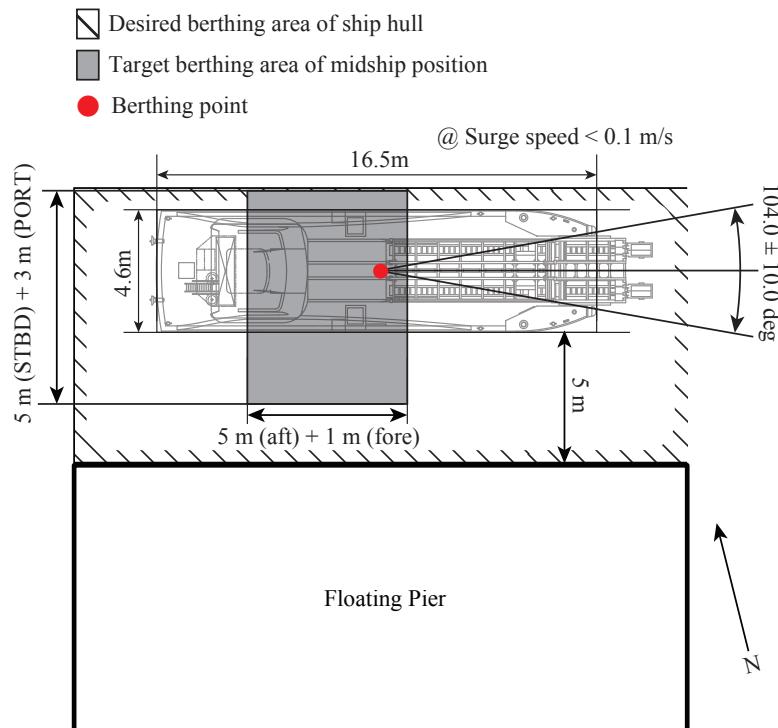


Figure 5.2 Terminal condition of automatic berthing control

PLC システムとの通信頻度は、ノートパソコンと搭載 PLC システム間の安定したシリアル通信が可能な最小間隔を考慮して、270ms に設定した。これより短い通信間隔の場合、シリアル通信のエラーが発生した。数値シミュレーション、実船実験とともに、自動着桟制御の終了条件を Fig. 5.2 に示すように定義した。着桟の終了条件は、(a. 位置条件) ミッドシップ位置が $8 [m] \times 6 [m]$ の矩形領域（図中グレーで囲まれた領域）内にあること、(b. 方位角条件) 船首方位角が浮桟橋と平行になる角度 ($104 [\text{deg}]$) に対して $-10 [\text{deg}]$ から $+10 [\text{deg}]$ 以内の位置にあること、(c. 速度条件) 船体の前後方向速度が $0.1 [\text{m/s}]$ 以内になることの 3 条件から構成されている。着桟終了条件のミッドシップ位置は、図に示す目標とする着桟エリア内に船が入るように設定されており、着桟ポイント（赤点）から浮桟橋までの距離は、船から桟橋に係留ロープを取ることができる距離として、今回は $5 [\text{m}]$ を想定して設定した。本研究では、この 3 つの終了条件を満たした場合を自動着桟の成功判定の基準として用いる。

5.3 数値計算による検証

本節では、自動着棧の基本的な性能を検証するために、自動着棧制御に関する数値シミュレーション結果を示す。

5.3.1 経路追従制御および風外乱補償制御の検証

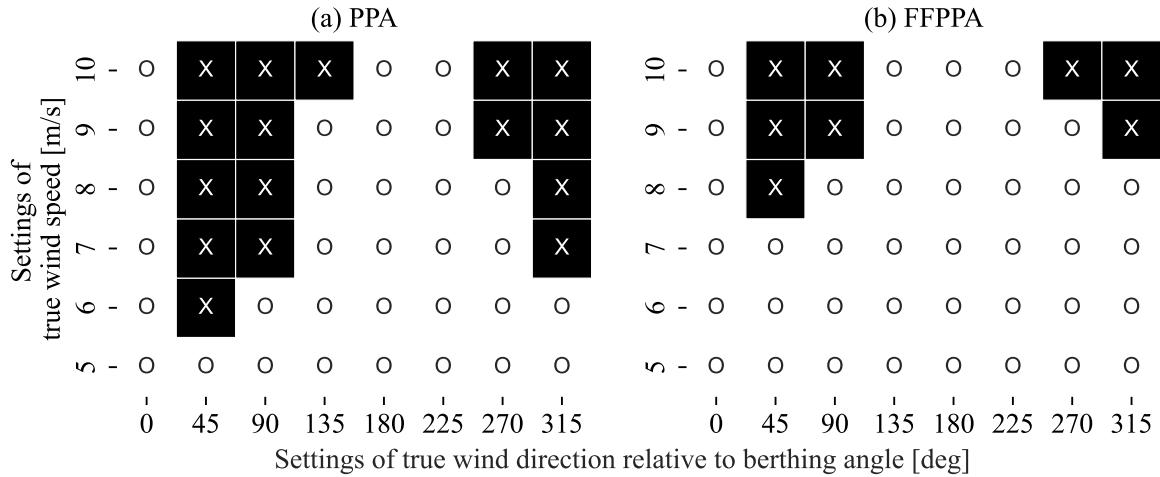


Figure 5.3 Results of automatic berthing control under constant wind conditions using the PPA and the FFPPA algorithm in numerical simulation. The white "O" cells represent successful berthing and the black "X" cells represent unsuccessful berthing. The initial ship's heading angle was 104 [deg] for all simulations.

FFPPA アルゴリズムを用いた自動着棧制御の数値シミュレーションの結果を示す。比較のため、PPA アルゴリズムによるシミュレーションも実施した。まず、定風下で真風速と風向の組み合わせを変えてシミュレーションを行い、その結果を Fig. 5.3に示す。シミュレーションの結果、PPA アルゴリズムは真の風速が 5 [m/s] のときのみ全ての風向を扱うことができ、FFPPA アルゴリズムは真の風速が 7 [m/s] までは全ての風向を扱うことができる事が確認できる。真の風速が 7 [m/s] 以上の場合、FFPPA アルゴリズムは一部の風向で設定した終了条件を満たすことができなかったが、PPA アルゴリズムと比較してより多くの真の風速と風向の範囲で終了条件を満足した。また、変動風に対しても同様のシミュレーションを行った。変動風でのランダム性の効果を定性的に評価するため、真の風速と風向きの組み合わせごとに各 10 回のシミュレーションを行い、成功率を算出した。その結果を Fig. 5.4に示す。変動風シミュレーションでは、定常風シミュレーションに比べ、両アルゴリズムによる成功率が低くなっている。定常風シミュレーションと同様に、FFPPA アルゴリズムは PPA アルゴリズムよりも多くの風条件で自動着棧に成功した。また、変動風下の自動着棧制御の数値シミュレーションにおける経路偏差を Fig. 5.5に示す。前章で述べた制御機の設計の狙い通り、FFPPA アルゴリズムが風外乱下における経路偏差を低減できていることがわかる。特に、風向が 135 [deg] と -45 [deg] の例では、PPA アルゴリズムと FFPPA

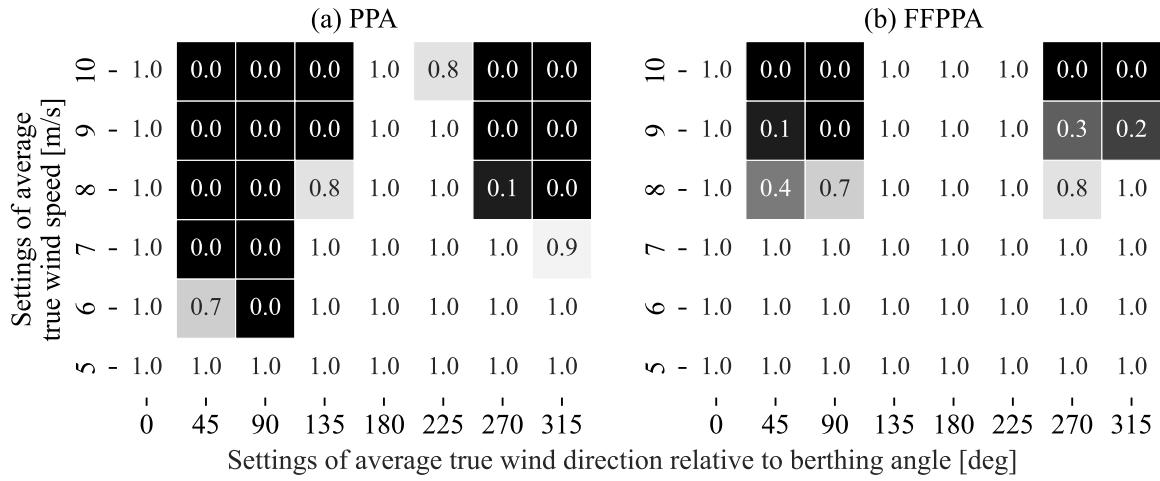


Figure 5.4 Success rate of automatic berthing control under variable wind conditions using the PPA and the FFPPA algorithm in numerical simulation. Each case was simulated ten times to calculate the success rate. The initial ship's heading angle was 104 [deg] for all simulations. The winds were simulated using Eq. (A.16) for the speed and Eq. (A.17) for the direction of a true wind. The average true wind speed was determined as the scale parameter of the Weibull distribution calculated using Eq. (A.18). The variance of the wind direction was set at 30 deg based on the observation around the test sea area.

アルゴリズムの差が大きく、これは経路をたどる際に船体が受ける横風の影響であると推察される。逆に、針路と平行に風が吹く 45 [deg] や -135 [deg] の例では、経路の大部分で両アルゴリズムの差は小さいが、45 [deg] の例では浮桟橋直前の経路偏差に差が生じていることが確認できる。

5.4 実船実験による検証

本節では、実験船神峰を用いた実船実験の結果を示す。

5.4.1 経路追従制御による自動着桟制御の検証

まず、経路追従制御の性能を検証するために、PPA 制御による自動着桟制御の基本的な結果を示す。Fig. 5.6 と Fig. 5.7 は、比較的風外乱の少ない状況での PPA 制御による自動着桟制御の結果を示している。風の影響下が少ないとおり Fig. 5.6 の状況では、精度良く経路追従制御ができていることがわかる。制御初期に生じた経路偏差も、時間の経過とともに現象しており、PPA 制御により経路偏差が抑えられている様子が確認できる。Fig. 5.7 に示される時系列データを見ると、船首方位角の振動が少なくスムーズに経路を追従すべく回頭できている事がわかる。また、船速についても設計通りに減速できていることがわかる。一方で、速度成分 u, v が細かく振動していることも確認できる。これは経路追従の過程で、船の横揺れの影響が含まれてしまっている事が考えられる。この成分を取り除くための方法としては、船体の動搖中心から測った GNSS のアンテナ高さを計測した上で、別途ロール角を計測するセンサを用いて横揺れによるアンテナの

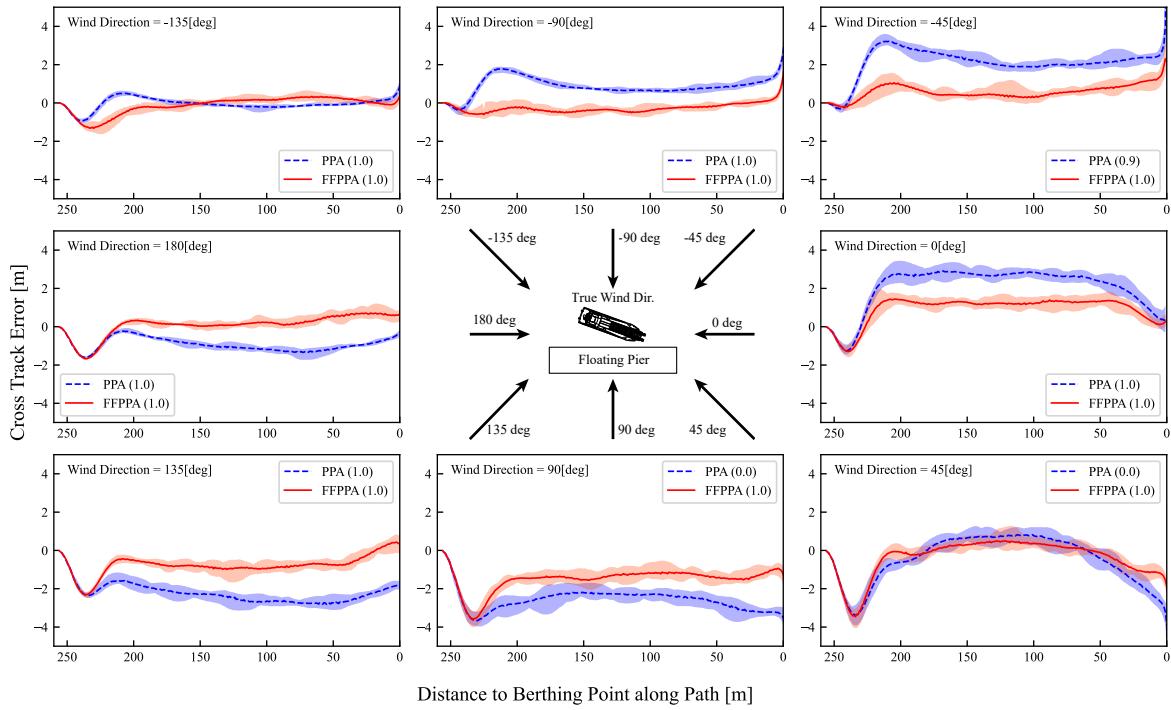


Figure 5.5 Cross track errors during automatic berthing control under variable winds at the average wind speed setting of 7 [m/s] in eight directions using PPA and the FFPPA algorithm in numerical simulation. The suffix of each legend represents the results of berthing control that (O) is success and (X) is a failure of automatic berthing. The winds were simulated using Eq. (A.16) for the speed and Eq. (A.17) for the direction of a true wind, where $\lambda = 14/\sqrt{\pi}$ [m/s] for true wind speed, $\sigma = \sqrt{30}$ [deg] for true wind direction. Each case was simulated ten times. Each line shows mean value and an area surrounding each line is filled between minimum and maximum values computed at every 1.0 m of the distance of the berthing point on a path. The initial ship's heading angle was 104 [deg] for all simulations. The cross track error is negative if the ship is left of the intended path.

移動分を差し引く方法が考えられる。今回は、横揺れ運動成分を計測するセンサを搭載しなかったため、これらの影響を含んだ値が制御で利用されている。ただし、制御則で利用される速度成分は主に前後方向の速度 u であるため、横揺れ成分の影響は相対的に小さく、制御結果に大きな影響はなかったものと推察される。

続く Fig. 5.8 と Fig. 5.9 は、風外乱の影響下にある状況での PPA 制御による自動着棧制御の実験結果を示したものである。このときの平均風速は 5.34 [m/s] であり、瞬間に 8.0 [m/s] を超す場面があった。この例では、風況が落ちている Fig. 5.6 の場合と比較して、経路偏差は大きくなっているが、その大きさが増大することなく経路追従していることがわかる。最終的に、着棧位置まで船体を制御しているが、一方で船首方位角は目標の 104 [deg] には到達しておらず棧橋側に寄っている。これは、回頭モード中に、船尾の右側から風を受けたことにより、船首方位角制御がオーバーシュートしたものと捉えられる。Fig. 5.9 に示される時系列データ

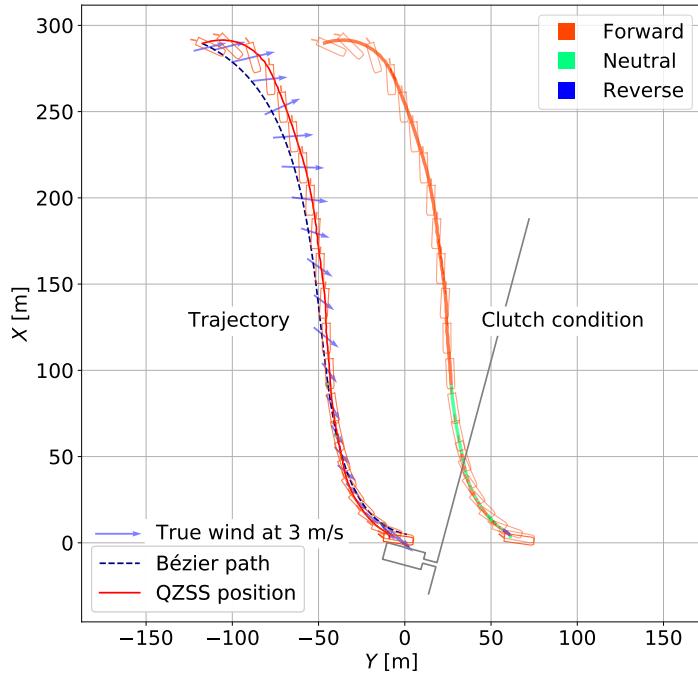


Figure 5.6 Trajectory of automatic berthing control using the PPA in a calm wind

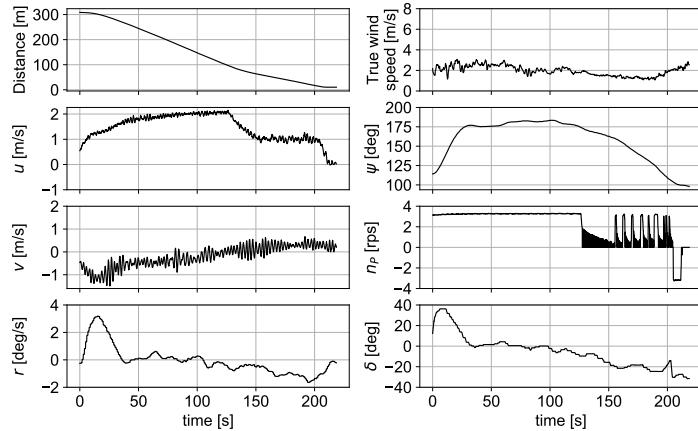


Figure 5.7 Measured values of the automatic berthing control shown in Fig. 5.6

を見ると、船首方位角の振動がFig. 5.6の結果と比較して、特に制御の前半の区間で大きくなっていることがわかる。ただし、最終的にその振動は抑えられており、低速となる回頭モードでは適切に方位角が制御されている。着桟の結果としては終端状態が設定した条件を満足しているものの、船首方位角は最終的に 112.7 [deg] であり余裕は小さかった。

5.4.2 風外乱補償制御の検証

Fig. 5.10とFig. 5.11は、提案する風外乱補償制御を導入した場合の自動着桟制御の実験結果である。Fig. 5.10とFig. 5.11に示すように、本実験は平均真風速 6.38 [m/s] のもとで行われた。制御

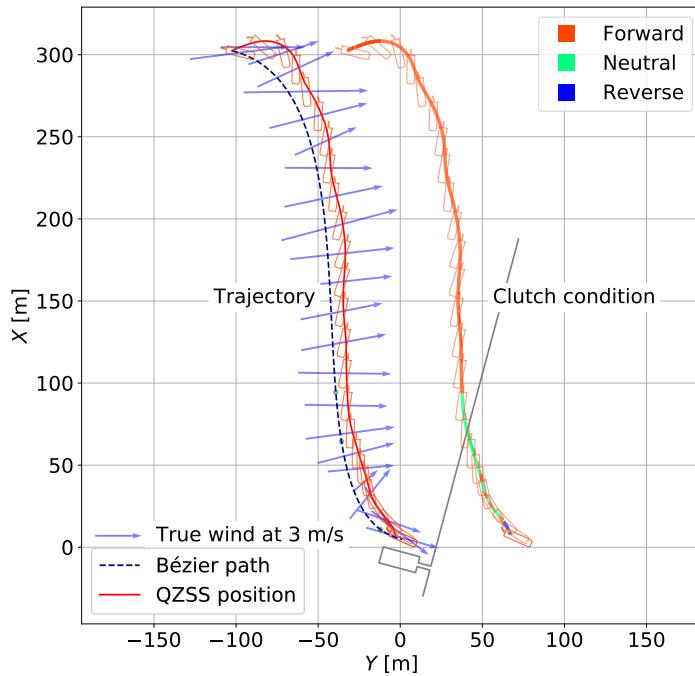


Figure 5.8 Trajectory of automatic berthing control using the PPA under strong wind disturbance

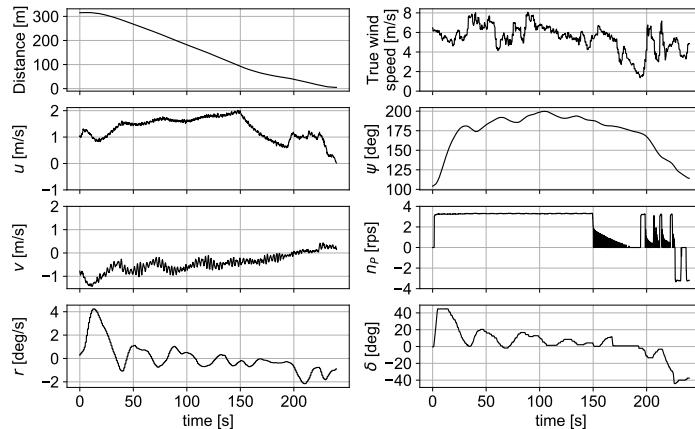


Figure 5.9 Measured values of the automatic berthing control shown in Fig. 5.8

初期には節制した経路偏差は徐々に減少し、浮桟橋から 50 [m] の距離で制御精度が要求される回頭前までに偏差はほぼ解消された。比較のため、近い風外乱の条件に対してフィードフォワード制御を行わない PPA 制御での自動着桟の実験結果を Fig. 5.12 と Fig. 5.13 に示す。こちらの実験では平均真風速が 6.34 [m/s] であったが、この風速では風外乱のフィードフォワード制御を行わないと一度発生した経路ずれを低減することはできなかった。両ケースの自動着桟制御の最終状態を Table 5.1 に示す。PPA アルゴリズムの例では、最終的な中船尾位置は着桟の終端条件を満たしているが、最終的な方位角は目標着桟角度から大きくずれており終端条件を満たしていない。

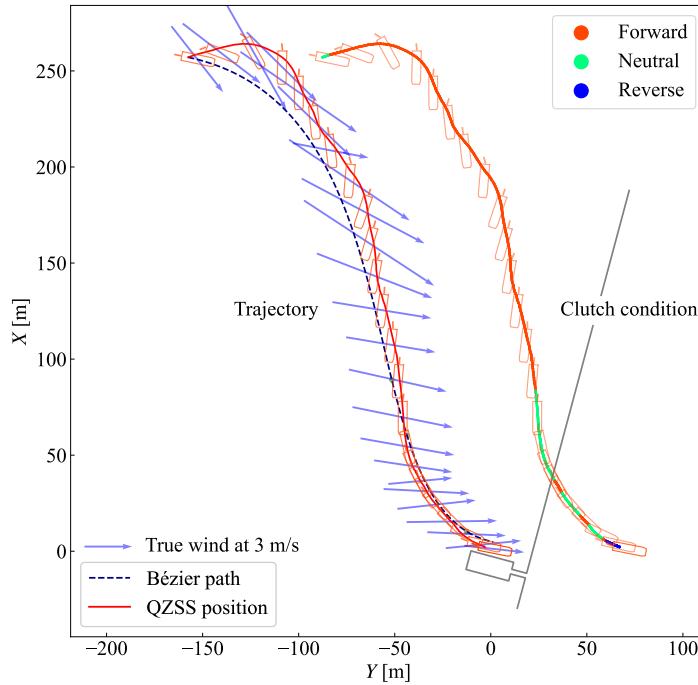


Figure 5.10 Trajectory of automatic berthing control using the FFPPA in wind disturbance environment

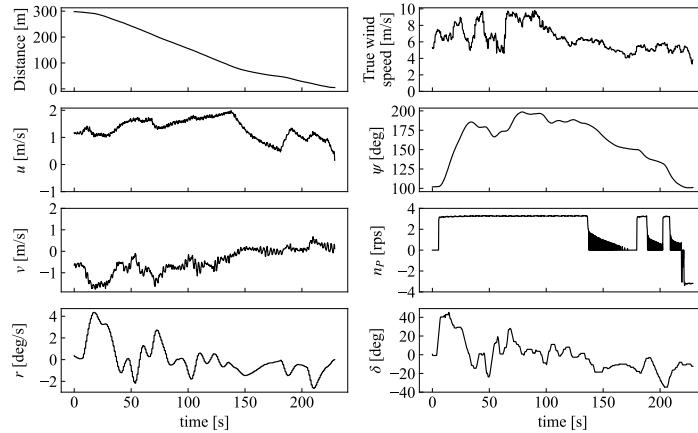


Figure 5.11 Measured values of the automatic berthing control shown in Fig. 5.12

一方、FFPPA アルゴリズムでは、すべての終局条件を満たしている。

5.5 その他の検証

5.5.1 目標方位角の補正に関する検証と考察

3.3.4で導入した補正角 α_{add} の効果について、実践試験により検証を行った。ここでは、桟橋への接近時の方位角を調整するために補正角 α_{add} を導入し、その角度を $-5, 0, 5$ [deg] と変えて経

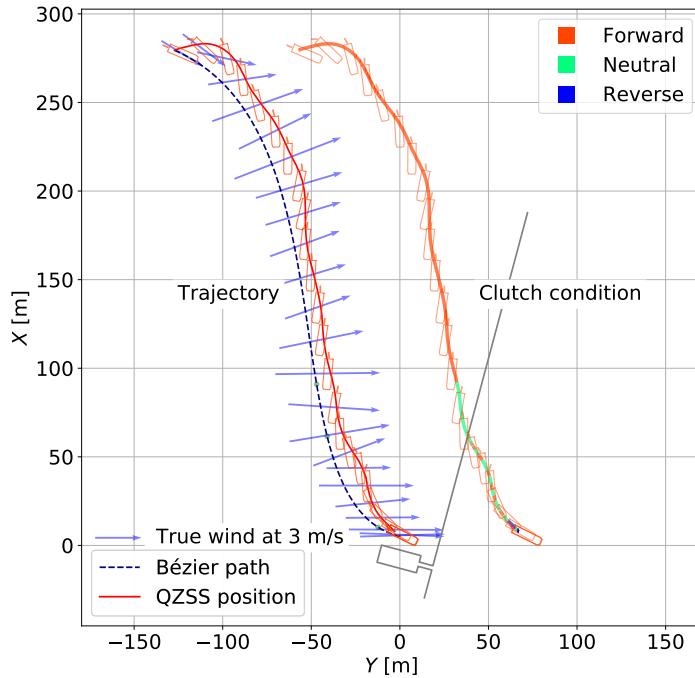


Figure 5.12 Trajectory of automatic berthing control using the PPA in wind disturbance environment

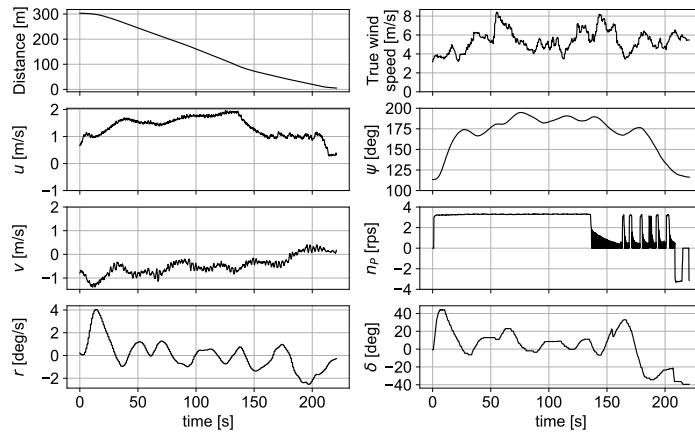


Figure 5.13 Measured values of the automatic berthing control shown in Fig. 5.12

路追従制御を実施した。Fig. 5.14に、その結果を示す。なお、本実験は、風外乱補償制御を導入していないPPA制御によって実施された結果である。Fig. 5.14では、補正角の影響が現れる桟橋周辺における経路追従誤差と補正角を除いた真の目標点の方位の時系列が示されている。 α_{add} を大きくすることで、着棧位置からの距離が小さい場合に α_T を大きくすることができる。特に、 $\alpha_{add}=-5$ [deg]のとき、桟橋への接近中のほとんどの時間で α_T が正となる。これは、船首方位が桟橋に平行もしくは遠ざかる向きの状態で着棧位置へ接近していることを意味する。一方、 α_{add} の変化による終端方位角の大きな差はない。したがって、この方法は、経路追従中の航跡を大き

Table 5.1 Final state of automatic berthing control of FFPPA and PPA

Algorithm	X [m]	Y [m]	Heading angle [deg]	Surge vel. [m/s]
FFPPA	1.01	2.00	102.1 (-1.9)	0.046 (< 0.1)
PPA	0.180	6.15	120.4 (+16.4)	0.463 (> 0.1)

く変えずに桟橋に接近する際の方位角を安全側に制御するのに有効であるといえる。

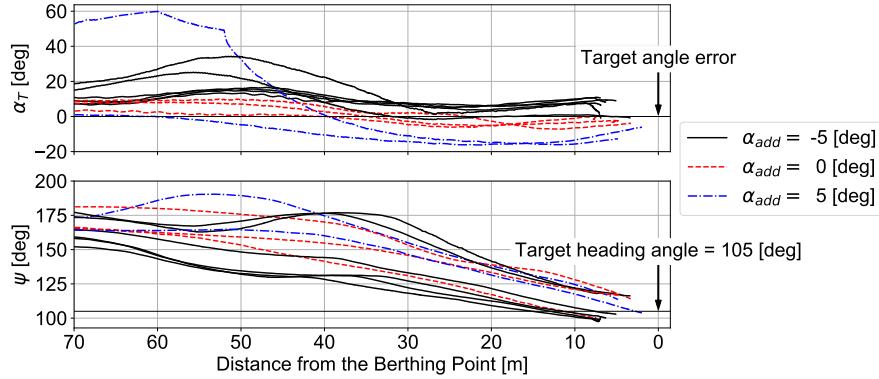


Figure 5.14 Comparison of time-series of the heading angle and α_T during automatic berthing control when changing α_{add} . Note that each value of α_T in the figure doesn't include α_{add} .

5.5.2 助走区間を設けた経路計画に関する検証

3.3.2.2において導入した、制御初期の経路に設けられた直線の助走区間による経路追従の効果を数値シミュレーションにより検証した。まず、助走区間の長さを変えた3つの風条件（無風、向かい風、追い風）でシミュレーションを行い、滑走路の設定が制御開始直後の経路偏差の抑制に与える影響について検討した。シミュレーション結果をFig. 5.15に示す。いずれの場合も、助走区間の長さに関係なく、助走区間がない場合と比較して、制御初期の発生する経路偏差が明らかに抑制されていることがわかる。今回の検証では、20 [m] の長さの助走区間を設けたケースが最もオーバーシュートが小さく、その後の実船実験に使用された。

5.5.3 PC と PLC による自動着桟制御の比較

自動着桟システム AutoBerth PLC を使用した実船実験の結果を Fig. 5.17に示す。PLC で制御した場合も、ノートパソコンで動作する AutoBerth で動作する場合と変わらない精度の自動着桟制御が可能であることがわかる。今回の実験では、経路計画に助走区間を導入しているが、着桟経路に助走区間を導入していない従来の実験結果と比較して、制御開始直後の経路偏差が解消されていることが分かる。

また、PLC の導入により、制御シーケンスの安定性も向上した。Fig. 5.18は、PC で動作する AutoBerth と AutoBerth PLC の2つの実装を用いた自動着桟の制御ループの計算時間を比較して

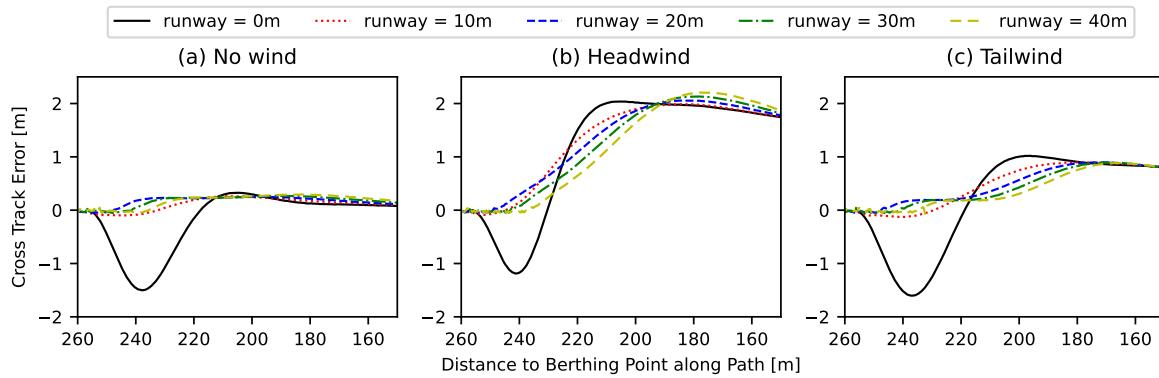


Figure 5.15 Cross track error of early stage of automatic berthing control with various straight runway distance of paths in numerical simulation. In all cases except (a), a constant wind was simulated and the true wind speed was 8.0 [m/s], the direction was 104 [deg] for (b) and 284 (=104+180) [deg] for (c). The initial ship's heading angle was 104 [deg] for all simulations. The cross track error is negative if the ship is left of the intended path.

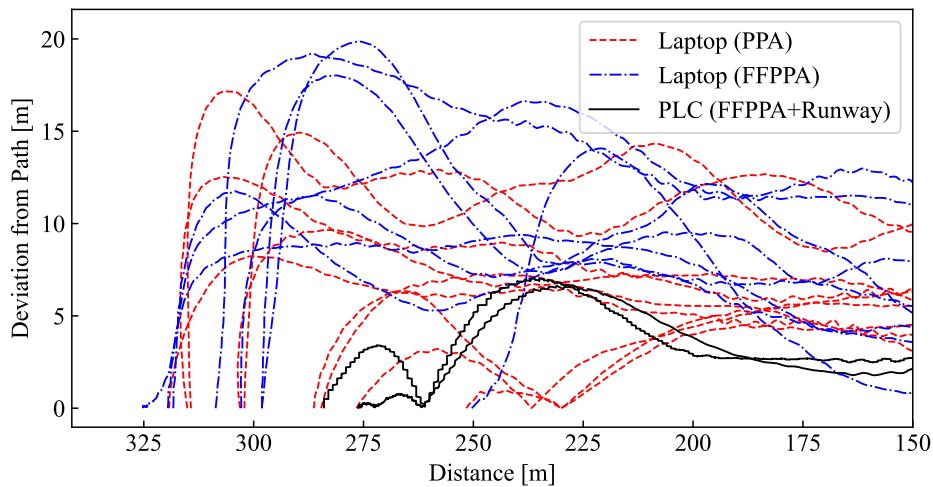


Figure 5.16 Cross track error of early stage of automatic berthing control with/without straight runway of paths.

いる。ノートPC上で動作する実装では、タイマによる制御を行わずに制御ループを回す実装では、主に可視化処理にかかる時間を主要因として計算時間が時間の経過と共に増加した（赤点線）。タイマを導入することでノートPC上で動作する実装での制御間隔は安定したが、シリアル通信間隔の制約から270 [ms]を下回ることはできなかった。タイマを導入しても、ノートパソコンでの実行では制御間隔にばらつきが発生し、制御間隔が800 [ms]近くまで低下する瞬間があった。対して、AutoBerth PLCは、船上のメインPLCシステムとEthernetで接続され、制御周期をタイマで制御している。さらにAutoBerth PLCは船内のメインPLCシステムと高速なEthernetで接続されており、結果として最短の計算間隔と安定した性能を発揮することができることを確認した。

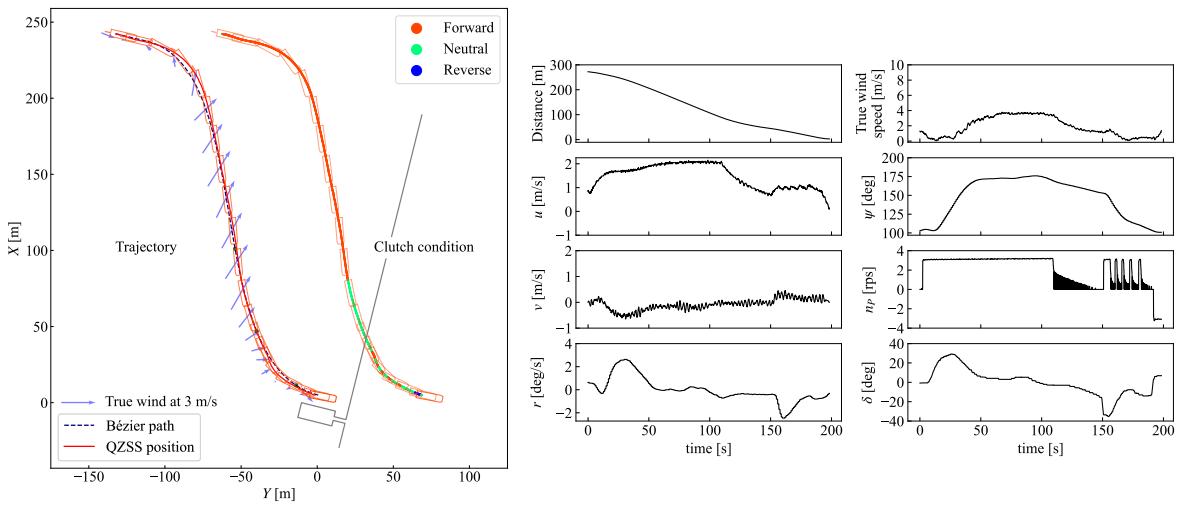


Figure 5.17 Trajectory and measured values of automatic berthing control with Auto-Berth PLC in wind disturbance environment

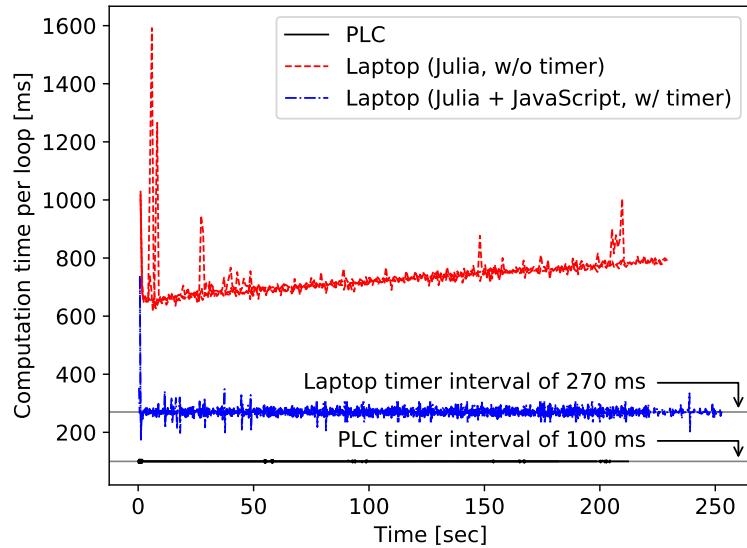


Figure 5.18 Comparison of computation time per loop. Each case includes measurements of three trials respectively.

5.6 結言

本章では、自動着棧制御アルゴリズムの検証を行った。検証を行うにあたり、自動着棧制御における満足すべき終端条件を示した。最初に、数値シミュレーションにより、定常風と変動風下におけるPPA制御アルゴリズムとFFPPA制御アルゴリズムによる自動着棧の結果を示した。PPA制御は、5 [m/s]以下の風速においては、安定的に自動着棧を行った。対してFFPPA制御は、7 [m/s]

の風速まで自動着棧の終端条件を満足することができた。また、一部の風向条件においては、より大きな風速であっても終端条件を安定して満足する結果となった。終端条件でなく、自動着棧制御中の経路偏差についても比較を行った。ここでも、変動風下において、FFPPA 制御の結果がより小さな経路偏差で自動着棧を行った。続いて、実船による比較も行った。まず PPA 制御について、比較的風況が穏やかな条件と、風のある状況において自動着棧を実施した。いずれにおいても終端条件は満足したが、風のある状況では、着棧直前まで経路偏差が小さくならず、また終端の船首方位角も目標からの偏差が大きくなかった。FFPPA 制御については、近い風速条件において PPA 制御との比較を行った。PPA 制御に比べて、FFPPA 制御の結果では、発生した経路偏差を徐々に低減していく様子が見られた。数値シミュレーションおよび実船実験の結果から、導入した風外乱補償制御の有効性を示すことが出来た。

また、目標方位角の補正に関する検証では、補正角を 3 つの値に設定して実船実験を行い、アプローチ時の航跡を大きく変えることなく、安全な船首方位角でのアプローチ操船の実現に有効であることを示した。同様に助走区間にに関する検証では、数値シミュレーションおよび実船実験によって、自動着棧の開始後に発生する経路偏差の発生を抑えることを確認した。最後に、PC と PLC による自動着棧制御の比較を行い、PLC がより安定的で高速な制御周期を実現する結果を示した。今後、より制御性能を向上させるために、外乱として潮流への対応や、速度制御の高度化が課題である。

第 6 章

自動着桟制御に関するその他の検討

6.1 安全な自動着桟システム

本研究で提案する風外乱補償制御は、主に舵の操作を工夫することで風外乱に対するロバスト制御を実現するものである。ただし、実験では安全のため主機回転数をアイドリングに固定し、船速の制御はクラッチ操作のみに限定している。制御工学的には、エンジン回転数を上げることで制御余裕が生まれ、コントローラの外乱に対するロバスト性が向上が見込める。実際の船舶運航では、風が強い時に着桟するための操船を行う際に、主機回転数を上げて方位角を制御することがある。より人間の操船に近い性能を持つ着桟自動制御装置を実現するためには、より高いエンジン回転数での制御を考慮する必要がある。一方で、制御装置の設計だけでなく、十分な安全対策も必要であり、これらの機能は、今後 MASS の他のシステムとの組み合わせで実現することと合わせて将来の研究課題である。

また、本研究では、ODD から逸脱した場合の対応として、画面表示や音声による乗組員への警告を行う機能を実装した。将来、船上のフォールバック要員が少ない場合や、船橋から離れている場合、完全自動化された船舶が想定される場合には、ODD から逸脱した場合にシステムが自律的な MRM を行い、システムが船舶を安全な状態を可能な限り維持するように設計される必要がある。また、MASS の安全な運用のためには、港湾形状や船舶を含む周辺環境の認識、センサデータの健全性や完全性監視も必要である。提案の自動着桟システムは、前章で述べたように、PPA 制御のアプローチ角度の補正と経路計画のアプローチ角度を適切に設計することにより、船舶操縦者が自動着桟制御を停止する判断をしやすくするように設計した。これは、急な角度で桟橋に接近し、あるいは浅い角度でも桟橋へ向かうの針路をギリギリまで保ってしまい桟橋の手前で急旋回するような場合など、操船者に不安を与えることになるような制御を避ける狙いがある。このような問題のある操船は、MRM の設計にも不向きである。緊急時に MRM を実行できるような余裕を持ったコントローラを設計することが重要であり、事故の直前まで制御が正しく動作しているかを判断しにくい操船は、同時に、旋回角度の不足、システムの不具合などにより、自動着桟が失敗したと判断した場合に、MRM が安全に停止して自動着桟制御に戻ることができるような余裕がないことになる。

本システムでは、自動着桟制御を行う乗組員が、自動生成された着桟経路が適切であることを制御を開始する前に確認することを想定している。今後、より高いレベルの MASS で自動着桟を実現する場合、周囲の環境に合わせた着桟経路を自動的に計画することが必要になる。そのため

には、静的情報として高精度な桟橋周辺地図や水深情報、動的情報として他船や障害物の検知システムが必要である。障害物を考慮した経路計画については、LiDAR を用いた手法が有効な解決策の一つになると考える。

6.2 経路追従制御に関する考察

本研究では、基本的な経路追従制御則である PPA 制御とこれに風外乱補償制御を組み合わせた FFPPA について、数値シミュレーションおよび実船実験による検証を実施した。一方で、経路追従の制御則には改良の余地があると考えられる。本節では、開発の過程で検討された経路追従制御について、いくつかの改善方法を検討した結果を示す。結論としては、これらの制御則による改善は一部で効果を示したもの、パラメータ調整等の問題があり、最終的な採用には至らなかった。

6.2.1 曲線経路形状の考慮

本研究で用いられる着棧制御時に生成する計画経路は、Bézier 曲線などのようにパラメタライズされた曲線の場合や、hybrid-state A* のようにパラメタライズされていない場合があるが、基本的には点列として表現されている。また、一般的な TCS で利用されるようなレグと曲線経路によって構成された経路もある。いずれにせよ曲線経路を追従する際には、Pure pursuit では少し先を見ていくことで、助走区間のシミュレーションでみたように経路への追従性を高めているが、その前方注視距離の設定には実験等の試行錯誤的な調整の作業が必要になってくる。その要因のひとつは、経路の曲率や船速等に応じて本来変えるべき前方注視距離が固定されていることによる。もともとの Pure pursuit 法は、前方注視距離を移動体の速度に応じて変更することが提案されている。これは、車輪で移動する対象には有効であるが、一方で今回の PPA 制御のような自動操舵との組み合わせでは、オートパイロットのゲイン調整とのバランスを撮る必要があるため設計が難しくなるという課題がある。ここでは、前方注視距離のかわりに、フィードバック制御へ組み込む形で曲率応じて目標角速度を設定して性能の改善を図る。与えられた経路に対する曲率を数値的に計算する方法として、経路を構成する点列に対して点列間の角度変化を計算する方法や、点列の 2 次微分値を近似的に得る方法などいくつかの方法が知られている。本研究の経路計画を Bézier 曲線で表現しているため、その曲率はパラメータ化された方程式を用いて解析的に求めることが可能である。一方で、一般的な点列で表現される経路への適用を考える場合には、メンガー曲率 [1] を利用することが考えられる。メンガー曲率は、与えられた 3 点 p_1, p_2, p_3 を通る円に対応する曲率 κ として次のように与えられる：

$$\kappa = \frac{4A}{|p_1 - p_2||p_2 - p_3||p_3 - p_1|}. \quad (6.1)$$

ここで、 A は三角形 $p_1p_2p_3$ の面積である。さらに経路追従時に左右の旋回を区別して扱えるよう、符号付きの曲率を得る方法としては、 A の計算を次のように行う：

$$A = \frac{1}{2}\{(p_1 - p_2)_x(p_3 - p_2)_y - (p_1 - p_2)_y(p_3 - p_2)_x\} \quad (6.2)$$

ただし, $(\cdot)_x, (\cdot)_y$ は、それぞれベクトルの x, y 成分を取る操作を表す。船体が曲率の経路に沿って

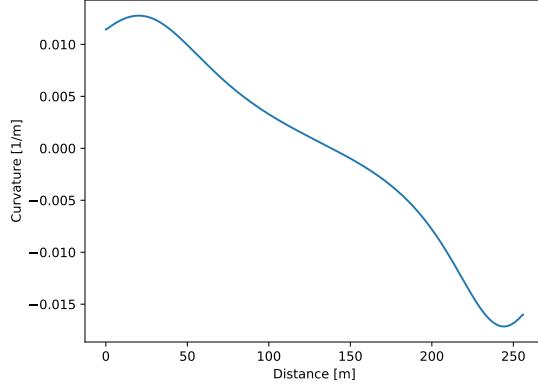


Figure 6.1 Curvature of the path created by a cubic Bezier curve.

運動しようとする場合、自船の surge 方向の速度 u が与えられると、幾何学的な関係から目標の回頭角速度 r_{des} は次の関係を満たす：

$$r_{des} = \kappa u. \quad (6.3)$$

この目標角速度 r_{des} を、船首方位角の微分制御に組み込むことで自船の回頭角速度を制御することができる：

$$\delta^* = \text{clip}(K_P[\alpha_T]_{-180^\circ, 180^\circ} - K_D(r - r_{des}), -45^\circ, 45^\circ). \quad (6.4)$$

6.2.2 斜航状態の考慮

風外乱等がある状況では、経路追従制御を行うと特に制御開始直後や回頭モードでは斜航状態で経路へ追従する場合が生じる。このとき、PPA 制御による経路追従では、目標方位に対する船首方位の差のフィードバック制御により舵角を計算するため、原則的に船首方位を目標点の方向に向けようと制御することになる。これは外乱をキャンセルするために斜航しながら経路を追従しようとする動きと衝突する。そこでこのような斜航状態が起きた状況に対応して、経路追従を行うために次のような重み付き方位角 ψ' を考える：

$$\psi' = \frac{u^2}{u^2 + v^2} \psi + \frac{v^2}{u^2 + v^2} \phi \quad (6.5)$$

ここで、 ϕ は、船の合成速度ベクトル U の方位角であり次のように定義される。

$$\phi = \psi + \beta + \text{atan2} \frac{v}{u} \quad (6.6)$$

Eq. (6.5) と Eq. (6.6) から重み付き方位 ψ' は次のように求められる。

$$\psi' = \psi + \frac{v^2}{U^2} \beta \quad (6.7)$$

重み付き方位 ψ' は、自船が斜航していないとき、つまり横方向の速度 v が小さいときは $\psi' \rightarrow \psi$ となるが、斜航状態ではその斜航角に応じて船首方位角 ψ からずれることになる。この重み付き方位を従来の船首方位の代わりに使用することで、直進時は従来どおりの動作をしつつ、斜航時には、斜航角の分だけ目標方位角への追従を抑える効果を期待できる。最終的に、指令舵角の計算は次のようになる。

$$\delta^* = \text{clip}\left(K_P \left[\alpha_T - \frac{v^2}{U^2} \beta \right]_{-180^\circ, 180^\circ} - K_D r, -45^\circ, 45^\circ\right). \quad (6.8)$$

6.2.3 経路偏差の直接的な補償

経路偏差を抑える手段として、追従時の経路偏差を直接フィードバックすることで、追従性能を向上させる方法について考える。PPA 制御では目標点の計算の際に、最初に船体位置に対して経路上の最接近点が計算される。今回はこの経路上の最接近点と船体ミッドシップ位置との距離を経路偏差 CTE として扱う。このとき、経路を進行方向に見るととき、船体が経路に左側にいるときを負、右側にいるときを正と定義する。Pure pursuit の前方注視距離を d_{LA} とするとき、目標の方方位角の偏差を次のように計算する：

$$\psi_{CTE} = \text{atan2} \frac{CTE}{d_{LA}} \quad (6.9)$$

これを船首方位角制御の目標方位に加えることで次のように制御則を得る。

$$\delta^* = \text{clip}(K_P[\alpha_T - \psi_{CTE}]_{-180^\circ, 180^\circ} - K_D r, -45^\circ, 45^\circ). \quad (6.10)$$

6.2.4 シミュレーションによる検討と考察

従来の FFPPA を基準として、着桟時の経路追従精度を改善するために、3 つの手法を追加して計 4 つの方法を比較した。比較手法は、FFPPA に経路偏差を導入した方法 (FFPPA-CTE)、FFPPA 法に重み付き方位を導入した方法 (FFPPA- β) と FFPPA 法に回頭角速度 r_{des} を導入した方法 (FFPPA- κ) である。これらの手法について、風の条件を変えて自動着桟制御時に発生する経路偏差を比較した。

まず最初に無風の条件下における基本性能を比較する。4 つの手法による自動着桟の数値シミュレーション結果を、Fig. 6.2 に示す。無風時では FFPPA と FFPPA- β はほとんど性能が変わらなかった。一方で、経路曲率を考慮した FFPPA- κ は曲線経路のより内側を追従するようになっており、結果として経路偏差は悪化する結果となった。経路偏差をフィードバックした FFPPA-CTE は、今回の 4 手法の中でもっとも経路偏差が小さかった。特に桟橋まで 200~250 [m] の区間や、50 [m] 以下の区間の曲線経路の部分では、経路偏差を抑えられている。

続いて、風がある場合の検証として、定常風と変動風の例を示す。まず定常風下における 4 手法の自動着桟の数値シミュレーションの結果を Fig. 6.3 に示す。図に示すように、桟橋の方向を基準として、8 方向からの風に対してシミュレーションを行った。無風時のシミュレーション結果と同様に、FFPPA と FFPPA- β はほとんどの区間で、経路偏差に関して差がみられなかった。た

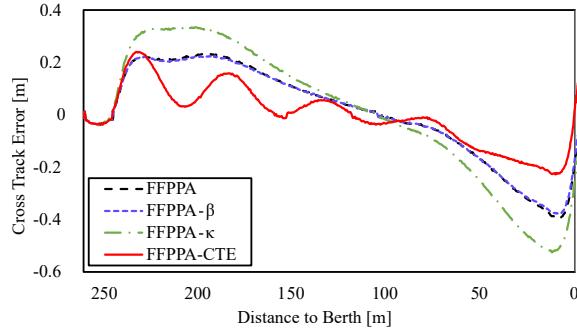


Figure 6.2 Cross track error during automatic berthing control (no wind)

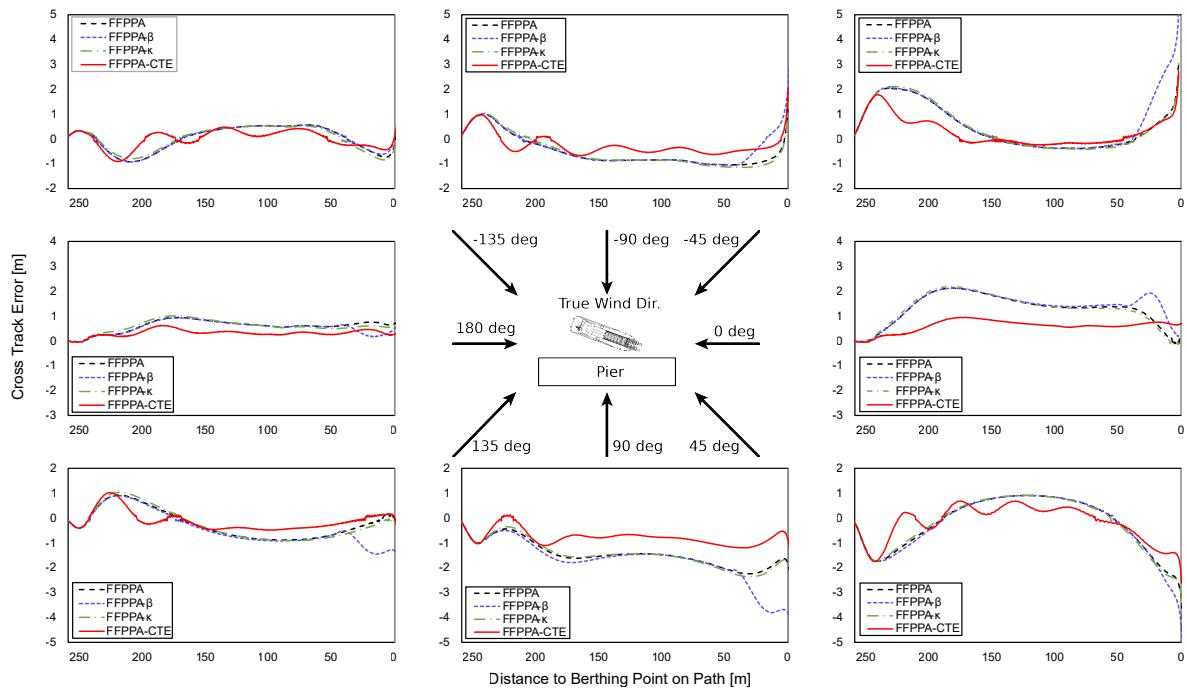


Figure 6.3 Cross track error during automatic berthing control for wind in 8 directions (constant wind, the true wind speed is 8.0 [m/s], the inlet distance of a Bézier path is 20 [m].)

だし、着桟直前の距離 50 [m] 以下の範囲では、一部のケースで FFPPA と FFPPA- β /FFPPA- κ と経路偏差の変化に差が見られた。その差は風向によっても異なり、風向が桟橋に対して 180 度の場合を除いて、FFPPA- β /FFPPA- κ の方が経路偏差が抑えられている。4 つの手法と比較して、すべての風向きで FFPPA-CTE は大きく経路偏差を改善していることが確認された。全ケースの中で、桟橋に対して +45 度の場合が最も成功率が低く、桟橋までの距離が 0 になったときの速度 u は、FFPPA-CTE が 0.083 [m/s] で目標の 0.1 [m/s] 以下であったのに対し、FFPPA が 0.18 [m/s]、FFPPA- β が 0.19 [m/s]、FFPPA- κ が 0.21 [m/s] となっている。これは着桟位置において、船速が 0.1 [m/s] 以下かつ船首方位角が桟橋の方位 (104.0 [deg]) の ± 10 [deg] 以下を着桟完了とする基準

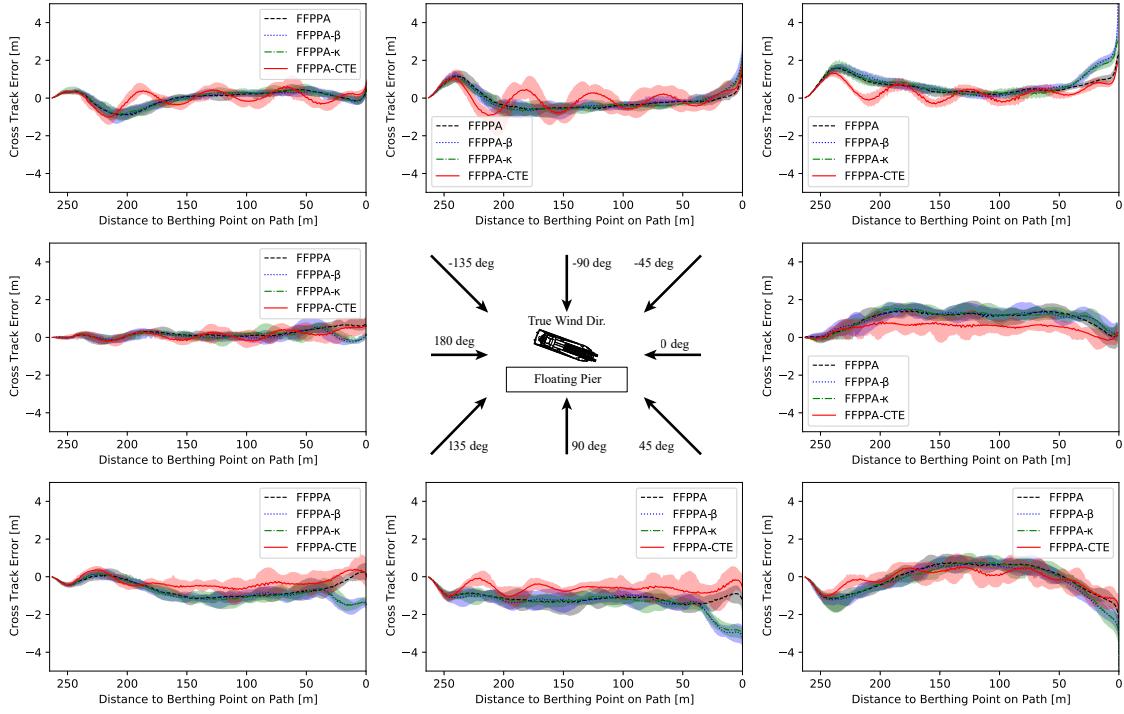


Figure 6.4 Cross track error during automatic berthing control for wind in 8 directions (Random wind with $\lambda = 14\sqrt{\pi}$ [m/s] for true wind speed, $\sigma = \sqrt{30}$ [deg] for true wind direction, inlet distance of a Bézier path is 20 [m].). 10 trials for each wind direction. Each line shows mean value and an area surrounding each line is filled between minimum and maximum values computed at every 1.0 [m] of the distance of the berthing point on a path.

を満たしていない。特に FFPPA- β に関しては、他に風向きが桟橋に対して 90 度, -45 度の場合にも、着桟の条件を満たさなかった。これ以外の風向きと制御手法の組み合わせでは、この基準を満たしている。

最後に変動風でシミュレーションを実施した結果を Fig. 6.4 に示す。変動風の生成には、真風速が 8.0 [m/s] 程度になるようにスケールパラメータを定めた。風向の標準偏差は、過去の因島マリーナ周辺の実測値を参考に $\sqrt{30}$ [deg] とした。ここでも、多くの場合で FFPPA-CTE の経路偏差が最も改善が見られた。平均風向が -45, -90, -135 [deg] の桟橋に向かって風が吹いている例では、FFPPA-CTE の経路偏差の変化に幅がある。これは、FFPPA-CTE が経路偏差が発生してから補償する比例制御に由来する動きであると推察される。このような経路偏差の振動を抑えるためには、一つに積分制御の導入が考えられるが、着桟時の船首方位は常に変化しているため、経路偏差の積分制御では各時刻で船が風を受けた方向を考慮する必要があると考える。

以上の通り、無風時と風のある状況で、FFPPA とその改良である 3 つの手法を比較した。いくつかのシミュレーションの結果から、FFPPA-CTE は改善が見られるケースが一番多かったが、一方で実際の環境に近い変動風下の自動着桟の数値シミュレーションでは、経路偏差の補償制御に由来すると推察される制御の振動が見られた。これを抑える方法を考えることが必要であるが、

元の FFPPA と比べて制御の自由度が増えることになり、ゲイン等の制御器の調整が難しくなると考えられる。

6.3 結言

本章では、これまでの議論を踏まえ、自動着棧制御に関するその他の考察を示した。まず、自動運航船に求められる機能として、安全性について改めて述べた。その中で、これまでの成果がどのような形で安全性に寄与するのかを整理した。さらに、経路追従制御に関して、第3章で述べた FFPPA 制御をベースに3つの手法を示した。これらの手法は、数値計算により無風、定常風、変動風の条件において自動着棧制御時の経路偏差を比較した。この中で、経路偏差を直接補償する制御手法が最も性能の改善が見られた。

第6章の参考文献

- [1] J.C. Leger, Menger curvature and rectifiability, Annals of Mathematics **149**(3), 831 (1999)

第 7 章

結論

本論では、自動運航船の技術のうち着桟操船の自動化に焦点を当てて、着桟制御および LiDAR を用いたセンシング技術に関する研究を実施した。1960 年代から始まった自動運航船の開発は、現在、IMO での議論が積極的に進められる段階に来ており、単純なアルゴリズムの性能のみならず、システムとしての安全性や利便性、規則上の扱いを考えてゆく必要がある。本論が取り扱った研究も、アルゴリズムの性能を求めるのみならず、人とシステムの関係において、安全に自動運航、特に着桟操船の自動化による支援を対象とした。自動運航船は非常に複雑なシステムであり、自動着桟に絞っても、周囲環境の状況認識、経路計画、経路追従、緊急時のヒューマンフォールバックや MRM といった機能で構成され、またシステムに接続される周辺のシステムとの連携等も検討する必要がある。本論では、着桟制御に注目し、自動運航船の全体システムは取り扱わなかつたが、通常の航行や避航操船等の他の操船モードとの切り替えなども実際のシステムでは考慮していくかなくてはならない。一部は、すでに検証を進めているが、自動運航船に関する業界と協力しつつ早期にシステムの具体的な姿を示すことが必要であると考える。将来的に求められるシステムの認証においては、これらの要素の検討に加え、リスク分析をベースに安全性を示すことが求められると予想される。安全性を担保するためには、性能の向上に加えて、様々な安全対策や冗長化を講じていく必要があると考えられる。本論では、このような観点の元、実用的な制御アルゴリズムを示すとともに、システムへの実装、ODD やアラート、MRM の検討、LiDAR を用いたセンサ系の冗長化について体系的に研究を実施した。本章では、これまでに示した研究を総括し、今後の課題について述べる。

第 1 章では、自動運航船の歴史を概観するとともに、本論の研究背景を述べた。自動運航船の社会実装の動きは年々強まっており、アルゴリズムだけでなくシステムの実装を踏まえた研究が求められている状況のもとで、自動着桟に対象を定め、アルゴリズムからシステムの実装、LiDAR の適用について研究を実施した。

第 2 章では、本研究に関連する先行研究について整理した。自動着桟の研究については、風外乱を扱ったものとそれ以外に分けて整理した。加えて、船舶のシステムの実装に関する事例についても整理した。LiDAR については、特に LiDAR SLAM による船舶の自己位置推定に関する研究を取り上げた。

第 3 章では、本論で用いた実験船、制御システム、アルゴリズム、実装した自動着桟システムについて述べた。実験船には、自動着桟に必要なセンサ系が搭載され、アクチュエータを含めて PLC を中心とした制御システムにより管理されている。提案システムを幅広い種類の船舶へと適用すべく、最も基本的な 1 軸 1 舵船を想定して設計を行った。自動着桟アルゴリズムは、着桟の

経路計画と経路追従制御により構成される。提案手法は、実用性を考慮して、計算が軽量でリアルタイムな動作が可能な形を取っている。経路追従制御は、Pure Pursuit アルゴリズムと PD 制御による Heading Controller により構成される PPA 制御法を提案し、必要なパラメータを最小限に抑えた。さらに、低速時に桟橋周辺で顕著となる風外乱の影響を軽減するために、船体に働く風圧力を推定するフィードフォワード制御による操舵を加えた 2 自由度制御器である FFPPA 制御法を提案した。この過程では、船体運動を線形の状態方程式として定式化し、船体に働く風圧力に対する相当舵角量を求める線形方程式を導出した。これらのアルゴリズムを実装するにあたり、開発時の効率と安定性の観点から、PC 上で動作する AutoBerth と PLC に実装された AutoBerth PLC を開発した。PC 上で動作する AutoBerth は、Web ベースの技術を用いたリッチな GUI を備えることで、運用中に制御システムが正常に動作しているか瞬時に判断可能な情報を提供する。また、計算部には高速な Julia 言語を採用することで、安定した制御を実現した。また、AutoBerth を含む、開発環境においては、PC 上で動作するシミュレーション環境、実船の PLC システムを接続したミニシミュレータを活用することで、船上における開発時間を最小化する効率的な開発フローを示した。一方で、AutoBerth PLC は、産業用のリアルタイムコントローラである PLC に、AutoBerth の機能を移植し、さらに PC では実現できない状態監視機能を実装した。本システムでは、ODD を設定することにより、自動着桟制御の実行時に、事前に想定した風況下や位置において安全にシステムが起動することを促すアラート機能を提供する。さらに、船上システムとの通信や接続先のシステムの相互状態監視機能を有することで、通信途絶時の MRM 実行機能を持っている。

第4章では、センサ系の冗長性の向上や自動着桟システムの高度化を目的に、LiDAR を用いた自動着桟のためのセンシング技術に関する3つの研究を実施した。1つ目は、障害物を回避する経路計画を実施する手法について数値計算による結果を示した。ここでは Hybrid-state A* アルゴリズムを、LiDAR で計測した桟橋周辺の岸壁の点群に対して適用し、前後進による切り替えを行う安全な経路が生成できることを示した。2つ目に、LiDAR SLAM による海上における点群地図作成と自己位置推定に関する研究成果を示した。LiDAR SLAM は、一般的に陸上の移動体に対して数多くの研究が行われてきたが、海上への適用を考えるといつか課題があった。本章では、LiDAR SLAM を海上で利用する際に、平面特徴量の欠如による Z 軸方向のドリフトと Roll, Pitch 方向の点群のスキャンマッチング誤差が発生する過程を述べた。この課題を解決するために、既存の SLAM アルゴリズムのスキャンマッチング計算において拘束条件を導入する手法を提案した。NDT および拘束条件を導入した LIO-SAM アルゴリズムを、実験船に搭載した LiDAR と 9 軸 IMU の計測値に適用し、後者のアルゴリズムにおいて 600 [m] の範囲で高精度な地図が生成できることを示した。また、QZSS や GPS の計測値と比較を行い、提案手法が高精度な QZSS と同等の精度で自己位置推定できることを示した。3つ目の研究では、システムへの実装として、LiDAR 点群を用いた岸壁検知システムを構築した。本システムは、PLC を用いて実装された。PLC の限られた計算資源を有効に活用すべく、RANSAC アルゴリズムを採用し、使用する点群データを抑えることでリアルタイム動作が可能な岸壁検知システムを構築した。ケミカルタンカーに本システムを搭載し、実際の岸壁を検知することで、岸壁と船首尾の距離を計測できることを確認した。

第5章では、自動着桟の数値シミュレーションおよび、実船の実験結果を示した。試験海域における右舷付の着桟制御を対象に、着桟制御における終端条件を示した。これに基づき、PPA 制御

と FFPPA 制御を、風の条件を変えてシミュレーションにより検証した。数値シミュレーションの結果から、PPA 制御は 5 [m/s] 以下の風速条件では安定的に制御できることを示すとともに、FFPPA 制御ではさらに多くの風条件で終端条件を満足する自動着棧制御を行った。実船実験においても、PPA 制御および FFPPA 制御による検証を行った。特に、FFPPA 制御は風の影響が大きい条件においても安定的に終端条件を満足する結果となった。この他に、安全な自動着棧を実施すべく、助走区間による経路追従時の経路偏差の低減や、アプローチ操船の最終段階における桟橋に対する船首方位角の改善を目的とした目標方位角の補正についても数値シミュレーションおよび実船実験による検証を実施した。また、PC と PLC の実装の違いによる制御システムの特性についても検証を行い、AutoBerth PLC のリアルタイム性能について、その優位性を示した。

第 6 章では、自動着棧制御に関するその他の検討について、これまでの研究を補完する議論をまとめた。まず、自動着棧システムの安全性について、これから MASS のシステムに求められる機能について述べた。その中で、これまでに述べた研究成果がどのように安全性に寄与するかについても述べた。さらに、経路追従制御の改良として、現時点を考えうるいくつかの改善案について、数値シミュレーションによる風外乱下における評価を行った。ここでは、経路の曲率、斜航状態の考慮、経路偏差の直接的な補償を行う 3 種類の手法を検討し、8 方向からの定常風および変動風を課した数値シミュレーションにより、オリジナルの FFPPA 制御と比較を行い、その特性を調べた。これらの中で経路偏差の直接的な補償を行う FFPPA-CTE が、比較的良好な結果を示したが、課題も見つかった。

本研究では、実用的な自動着棧システムに求められる機能の観点から、自動着棧制御アルゴリズム、自動着棧システム、開発フロー、安全対策、LiDAR によるセンサ系の冗長化に関する成果を示した。これまでの自動着棧制御を含む、自動運航船の研究においては、提案されたアルゴリズムの性能に対する問題点が指摘されることもなく、一方で実際のシステムとして実用化に繋がらない現実があった。自動運航船の実用化には、数多くの困難があるものの、一方で IMO の議論などを例に、少しづつ自動運航船のシステムに求められる機能や安全性の方向性が示されてきている。一方で、未だ具体的なアプローチについては研究例が少ないことも事実である。これから自動運航船の研究は、安全性についても考慮されたシステムとしての機能を提案する段階にあると言え、その点でアルゴリズムのみならず、状態監視やアラート、MRM などの安全対策、LiDAR による既存センサ系の冗長化に関する成果を示したことは、こうした状況に対する一つの指針を示すことができたといえる。これまでになかった複雑システムである自動運航船の認証には、リスク分析を軸とした非常に広範囲の安全性の証明が求められる流れにある。MASS Code の策定に対応したシステムの具体的な構成を提示するとともに、今後必要となる産業規格に対応するシステム全体の詳細な構成とその検査方法については今後の課題である。とはいって、今回示した ODD や MRM の実装、着棧の終端条件を含めた数値シミュレーション手順は、その足がかりになるものと考える。また、アルゴリズムの高度化についても依然として検討すべき課題が残っている。今回の試験海域では影響が少なかった潮流の影響や、速度制御の難しいより大型の船舶への適用についても今後、研究を進めていく必要がある。

謝辞

本研究を進めるにあたり、本学位論文は、大阪大学工学研究科地球総合工学専攻船舶海洋工学コースにおける後期博士課程の在学中に行った研究を中心にまとめたものです。はじめに、本研究の遂行に際して、多大なるご指導、ご鞭撻を賜りました。指導教官であります、大阪大学牧敦生教授に深甚なる謝意を表します。牧教授には、著者が就職前の前期博士課程から、ご退官された長谷川教授（当時）に代わって指導頂きました。就職後に、社会人として後期博士課程のお誘いを受け、学会等のオーガナイズドセッション主催やセミナー等を通して研究の発表、研究者との議論の場を広く与えていただきました。

また、本学位論文の多くは、所属する海上技術安全研究所において実施された研究を中心に構成しています。これらの研究は、平田宏一特別研究主幹が管理・整備していただいた実験船「神峰」および、船上システムなしには為し得ないものでした。平田特別研究主幹には、入所直後から、自動着棧の実船実験の機会を与えていただき、瀬戸内海の交通環境の中で操船やシステム開発等について非常に多くの学びを得ることができました。また、流体性能評価系の北川泰士主任研究員には、船体運動数学モデルの構築、操縦運動の実船計測等で様々な知見を頂きました。低速時の精緻な数値シミュレーションおよび、安定して動作する実船の制御システムの環境があったことは、本研究を遂行する上でこの上ない武器となりました。ここに深く謝意を表する次第です。

最後に、これまで私を支えてくださった両親に深く謝意を表します。

研究業績

本研究に関連する著者の研究論文は下記のとおりである。

査読付き論文（計 6 件）

Publication 1. Ryohei Sawada, Koichi Hirata, and Yasushi Kitagawa. Automatic berthing control under wind disturbances and its implementation in an embedded system. *Journal of Marine Science and Technology*, 2023. ISSN 1437-8213. doi: 10.1007/s00773-023-00934-9. URL <https://doi.org/10.1007/s00773-023-00934-9>.

Publication 2. Ryohei Sawada and Koichi Hirata. Mapping and localization for autonomous ship using LiDAR SLAM on the sea. *Journal of Marine Science and Technology*, 2023. ISSN 1437-8213. doi: 10.1007/s00773-023-00931-y. URL <https://doi.org/10.1007/s00773-023-00931-y>.

Publication 3. Yoshiki Miyauchi, Ryohei Sawada, Youhei Akimoto, Naoya Umeda, and Atsuo Maki. Optimization on planning of trajectory and control of autonomous berthing and unberthing for the realistic port geometry. *Ocean Engineering*, 245:110390, 2022. ISSN 0029-8018. doi: 10.1016/j.oceaneng.2021.110390. URL <https://linkinghub.elsevier.com/retrieve/pii/S0029801821016826>.

Publication 4. Atsuo Maki, Yuuki Maruyama, Leo Dostal, Masahiro Sakai, Ryohei Sawada, Kenji Sasa, and Naoya Umeda. Practical method for evaluating wind influence on autonomous ship operations. *Journal of Marine Science and Technology*, 27(4):1302–1313, 2022. ISSN 0948-4280, 1437-8213. doi: 10.1007/s00773-022-00901-w. URL <https://link.springer.com/10.1007/s00773-022-00901-w>.

Publication 5. Ryohei Sawada, Koichi Hirata, Yasushi Kitagawa, Eiko Saito, Michio Ueno, Katsuji Tanizawa, and Junji Fukuto. Path following algorithm application to automatic berthing control. *Journal of Marine Science and Technology*, 26(2):541–554, 2021. ISSN 0948-4280, 1437-8213. doi: 10.1007/s00773-020-00758-x. URL <https://link.springer.com/10.1007/s00773-020-00758-x>.

Publication 6. Ryohei Sawada, Keiji Sato, and Takahiro Majima. Automatic ship collision avoidance using deep reinforcement learning with LSTM in continuous action spaces. *Journal of Marine Science Technology*, 26:509–524, 2021. ISSN 0948-4280. doi: 10.1007/s00773-020-00755-0. URL <https://link.springer.com/article/10.1007/s00773-020-00755-0>

講演会論文（計 10 件）

Publication 7. 澤田涼平, 平田宏一. 自動着桟のための点群地図作成と自己位置推定. 日本船舶海洋工学会講演会論文集, (34):1–4, 05 2022.

Publication 8. 牧敦生, 丸山湧生, Leo Dostal, 笹健児, 澤田涼平, 脇田康希, 酒井政宏, 梅田直哉. 自動運航性能評価のための不規則風外乱の高速計算アルゴリズムの開発(第2報). 日本船舶海洋工学会講演会論文集, (34):13–19, 05 2022.

Publication 9. 宮内新喜, 澤田涼平, 和田翠星, 谷口拓也, 濱田暁, 小池弘顯, 脇田康希, 牧敦生. 自動運航船開発に適したシミュレータモデルとは. 日本船舶海洋工学会講演会論文集, (35):37–52, 2022.

Publication 10. 澤田涼平, 北川泰士, 平田宏一. 風外乱補償型経路追従制御による自動着桟制御. 日本船舶海洋工学会講演会論文集, (32):11–16, 2021.

Publication 11. 牧敦生, 丸山湧生, Leo Dostal, 酒井政宏, 澤田涼平, 笹健児, 梅田直哉. 自動運航性能評価のための不規則風外乱の高速計算アルゴリズムの開発. 日本船舶海洋工学会講演会論文集, (33): 65–75, 11 2021.

Publication 12. 澤田涼平, 平田宏一. 航行環境を考慮した着桟制御のための経路計画. 日本船舶海洋工学会講演会論文集, (32):37–40, 2021.

Publication 13. 澤田涼平, 平田宏一, 小林充, 仁木洋一, 佐藤圭二, 齊藤詠子, 宮崎恵子, 南真紀子, 國分健太郎. 自動運航船に向けた操船自動化システムの検証. 交通・物流部門大会講演論文集, 2021.30(0): TL3–2, 2021.

Publication 14. 澤田涼平, 平田宏一. 自動着桟における曲線経路追従制御. 日本船舶海洋工学会講演会論文集, (33):9–15, 11 2021.

Publication 15. 澤田涼平, 平田宏一. 航行環境を考慮した着桟制御のための経路計画. 日本船舶海洋工学会講演会論文集, (32):37–40, 2021.

Publication 16. 澤田涼平, 平田宏一, 北川泰士, 齊藤詠子, 上野道雄, 宮崎恵子, 谷澤克治, 福戸淳司. 経路追従制御による自動着桟操船システムの開発. 日本船舶海洋工学会講演会論文集, (30):43–49, 05 2020.

Appendix A

神峰の操縦運動数学モデル

本研究で用いた、神峰の操縦運動数学モデルについて本付録で説明する。このモデルは、様々な風条件下での神峰の自動着桟アルゴリズムの開発・評価に活用された。旋回などの一般的な操縦運動とは異なり、着桟操船にはプロペラ逆転を含む船速変化に対応したモデルが必要である。また、船上センサの項で述べたように、着桟操船には高精度の運動制御が必要であり、シミュレーションにおける運動予測の精度についてもこれに対応して十分に高いことが求められる。そこで、MMG (Mathematical Modeling Group) モデル [1] をベースに、船舶の操縦運動を高精度に予測できる動的モデルを開発した。運動モデルの流体力微係数などのパラメータは、神峰を用いた実船試験の結果に基づいて決定した。次節では、まず MMG に基づく操縦運動の運動方程式を説明する。そして、運動モデルのパラメータを調整する手順と風圧力の数値シミュレーション手順について簡単に説明する。なお、波や潮流については、これらの外乱による船体への影響が試験海域の桟橋周辺で十分に小さいと考えられるため、今回は取り扱わないこととした。

A.1 着桟操船時低速航行状態の操縦運動数学モデル

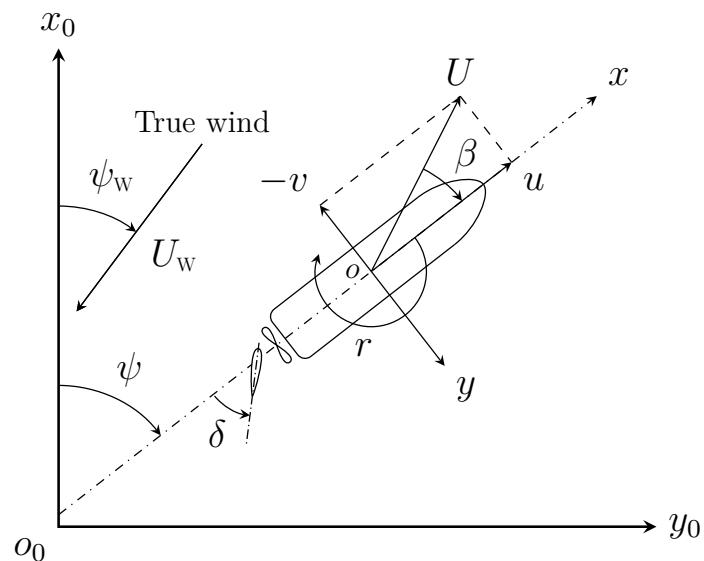


Figure A.1 Coordinate systems of ship motion and wind

本研究では、MMG モデルに基づく操縦運動の数学モデルを用いて、外乱下での自動着棧制御アルゴリズムに対する応答、特に風の影響を含めた低速時の船体運動を評価する。本論文で運動方程式を記述するために使用する座標系を Fig. A.1 に示す。各シンボルの表記は、文献 [1] に準ずる。Fig. A.1 に示すように、空間固定座標系を $o_0 - x_0y_0z_0$ と表記し、 x_0 を北、 y_0 を東向きに取る。船体固定座標系の原点 o を船体のミッドシップ位置に取る。 u , v , r はそれぞれ、ミッドシップ位置における surge 速度、sway 速度、回頭角速度である。船首方位角は ψ で表される。ミッドシップ位置での斜航角 β は $\beta = \tan^{-1}(-v/u)$ とし、合成速度を $U = \sqrt{u^2 + v^2}$ とする。3 自由度における船舶操縦運動の基礎方程式は、Eq. (A.1) のように表される。

$$\left. \begin{array}{l} (m + m_x)\dot{u} - (m + m_y)vr - x_Gmr^2 = X \\ (m + m_y)\dot{v} + (m + m_x)ur + x_Gm\dot{r} = Y \\ (I_{zG} + x_G^2m + J_z)\dot{r} + x_Gm(\dot{v} + ur) = N_m \end{array} \right\} \quad (\text{A.1})$$

ここで、surge, sway, yaw 方向の運動の付加質量と付加慣性モーメントをそれぞれ m_x , m_y , I_z と表記する。付加質量と付加慣性モーメントは、元良チャートの重回帰式に基づき決定される [2–4]。付加質量を除いたミッドシップ周りの船体に働く外力 X , Y , モーメント N_m は、標準的な MMG の作法に従って、物理的意味の観点から次の成分に分離して記述することができる。

$$\left. \begin{array}{l} X = X_H + X_R + X_P + X_A \\ Y = Y_H + Y_R + Y_P + Y_A \\ N = N_H + N_R + N_P + N_A \end{array} \right\} \quad (\text{A.2})$$

ここで、添え字の H , R , P , A はそれぞれ船体、舵、プロペラ、風圧力に働く力を意味する。バウスラスタによる力は、本論文では自動着棧制御で使用しないため省略した。

船体に作用する流体力学的な力の計算には、cross-flow drag の理論に基づき、芳村ら [5] によって提案された次のモデルを採用した。このモデルは、低速前進時の大きな斜航角を持つ運動の横方向と回頭方向の流体力学的な力を表現できる。このモデルの利点として、標準的な MMG モデル [1] と比較して、パラメータの数が少なくて済む。

$$\left. \begin{array}{l} X_H = \frac{1}{2}\rho Ld \left[\left\{ X'_{0F} + (X'_{0A} - X'_{0F}) \frac{|\beta|}{\pi} \right\} uU + \left(m'_y + X'_{vr} \right) L \cdot vr \right] \\ Y_H = \frac{1}{2}\rho Ld \left[Y'_v v |u| + Y'_r L \cdot ur - \frac{C_D}{L} \int_{-L/2}^{L/2} |v + C_{rY}rx| (v + C_{rY}rx) dx \right] \\ N_H = \frac{1}{2}\rho L^2 d \left[N'_v v |u| + N'_r L \cdot |u|r - \frac{C_D}{L^2} \int_{-L/2}^{-L/2} |v + C_{rN}rx| (v + C_{rN}rx) x dx \right] \end{array} \right\} \quad (\text{A.3})$$

ここで、 X'_{0F} と X'_{0A} はそれぞれ前進時と後進時の抵抗係数である。また、 X'_{vr} , Y'_v , Y'_r , N'_v , N'_r は流体力微分である。 C_D , C_{rY} , C_{rN} はそれぞれ Sway, Yaw に関する Cross-flow drag に関する修正係数を表す。前進時の抵抗係数は、プロペラ回転速度が 3.1 [rps] の実機での海上試験の結果に基づいて決定した。後進時の抵抗係数は、過去に実施された水槽試験結果のデータベースから決定された。

プロペラ正転および逆転時の舵とプロペラによる流体力は、北川ら [6] のモデルで表現し、プロペラ逆回転により発生する不均衡流を表現できるように次式の通りとした：

$$\left. \begin{array}{l} X_R = -(1 - t_R)F_N \sin \delta \\ Y_R = -(1 + a_H)F_N \cos \delta \\ N_R = -(x_R + a_H x_H)F_N \cos \delta, \end{array} \right\} \quad (\text{A.4})$$

$$\left. \begin{array}{l} X_P = (1 - t_P)\rho n^2 D^4 \cdot K_T(J) \\ Y_P = \begin{cases} 0, & \text{for } n \geq 0 \\ \rho n^2 D^4 \cdot Y_P^*, & \text{for } n < 0 \end{cases} \\ N_P = \begin{cases} 0, & \text{for } n \geq 0 \\ \rho n^2 D^4 L(x'_{PRR} Y_P^*), & \text{for } n < 0 \end{cases} \end{array} \right\} \quad (\text{A.5})$$

ここで、 $(1 - t_R)$ は操舵による推力減少係数、 a_H は舵力の修正係数、 x_H は付加的に作用する横力の作用点の船体縦方向 x の座標である。推力減少係数 $(1 - t_P)$ はプロペラ回転数 n の符号により異なる。プロペラ推力の計算に用いる $K_T - J$ 曲線は、MAU プロペラの性能推定近似式および B-Series プロペラの第 2~4 象限までの特性を推定するためのデータベースに基づき求めた。プロペラ前進定数 J は、次のように定義される：

$$J = \frac{u(1 - w_P)}{nD}. \quad (\text{A.6})$$

ここで、 D はプロペラの直径、 n はプロペラ回転数 [rps] である。プロペラ位置での伴流係数を w_P とする。プロペラの回転数については、本文で述べたように着桟位置近傍においては主機アイドリング状態でのクラッチの切り替えによる制御を想定している。そのため、実船の計測からクラッチの前進、ニュートラル、後進に対してそれぞれ 3.1, 0, -3.1 [rps] に対応するとし、今回はクラッチ切り替えの時間遅れは無視できるものとした。また、クラッチをニュートラルに設定した際の推力計算ではプロペラ回転数がゼロになるため、前進定数 J の計算を省略し、プロペラ推力 $T = X_P / (1 - t_P)$ に関してはゼロとした。そして、プロペラ推力 $T = X_P / (1 - t_P)$ は 0 として扱い、舵への縦方向の流入速度 $u_R = (1 - w_R)u$ は u_R の元の定義に基づいて計算されている。さらに、舵への前後方向有効流入速度 u_R を文献 [1, 7] の定義に基づいて $u_R = (1 - w_R)u$ とした。 $n < 0$ のときの推力減少係数 t_P は、次のように計算できる：

$$1 - t_P = C_{tp1} \cdot J_P + C_{tp0}, \quad (\text{A.7})$$

ここで、 $J_P := \frac{u}{n_P}$ は北川のモデルにおける見掛けの前進定数である。 C_{tp0} と C_{tp1} は J_P によって Table A.3 のよう設定し、これらの値はプロペラ逆転時の拘束模型試験のデータベースに基づいて決定された。舵直圧力 F_N は、次式で表される：

$$F_N = \frac{1}{2} \rho A_R U_R^2 C_N(\alpha_R). \quad (\text{A.8})$$

ここで、舵位置での合成流速度 U_R と舵に対する有効流入角 α_R は、以下のように表される：

$$\left. \begin{array}{l} U_R = \sqrt{u_R^2 + v_R^2} \\ \alpha_R = \delta - \delta_0 - \tan^{-1} \left(\frac{v_R}{u_R} \right) \end{array} \right\} \quad (\text{A.9})$$

ここで、 δ_0 は、舵直圧力の仮想的なゼロ舵角である。舵直圧力の係数 C_N は、次のように表される：

$$C_N(\alpha_R) = \begin{cases} C_{N0} \sin \alpha_R, & \text{for } 0 \leq |\alpha_R| < |\alpha_{Rstl}| \text{ or } (\pi - |\alpha_{Rstl}|) \leq |\alpha_R| \leq \pi \\ C_{N0} \sin(\text{sgn}(\alpha_R) \cdot \alpha_{Rstl}), & \text{for } |\alpha_{Rstl}| < |\alpha_R| < (\pi - |\alpha_{Rstl}|) \end{cases} \quad (\text{A.10})$$

ここで、 C_{N0} は舵直圧力の微係数であり、藤井の式 [8] を用いて決定された。また、舵のストール角は α_{Rstl} で示される。そして、舵への流入速度 u_R は、以下のように定義される：

$$u_R = \begin{cases} u(1 - w_R) \sqrt{\eta \cos^2 \alpha_S \cdot \left\{ 1 + k_x \frac{1 - w_P}{1 - w_R} \left(\sqrt{1 + \frac{8K_T}{\pi J^2}} - 1 \right) \right\}^2 + (1 - \eta)}, & \text{for } n \geq 0 \\ \text{sgn}(u_{RPRSQ}) \sqrt{(|u_{RPRSQ}|)}, & \text{for } n < 0 \end{cases} \quad (\text{A.11})$$

ここで、

$$\left. \begin{array}{l} u_{RPRSQ} = \eta \cdot u_{PRR1} |u_{PRR1}| + (1 - \eta) \cdot u_{PRR2} |u_{PRR2}| + C_{PRR} \cdot u |u| \\ u_{PRR1} = u(1 - w_R) + \cos \alpha_S \cdot n D_P k_{xPR} \sqrt{\frac{8|K_T|}{\pi}} \\ u_{PRR2} = u(1 - w_R) \end{array} \right\} \quad (\text{A.12})$$

操縦運動時の舵位置での航跡係数を w_R と表記する。 $\cos(\alpha_S)$ の項は、プロペラによる增速分を考慮した補正係数である。式中、 k_x は実験定数であり、プロペラ後流の増分率と呼ばれる。また、 η は舵全体の面積に対するプロペラ後流における有効舵面積の比である。ここで、横方向の有効流入速度 v_R は、次のように表される。

$$v_R = -\gamma_R(v + l_R r) \quad (\text{A.13})$$

ここで、 γ_R と l_R は整流係数を表す。

A. 1.1 風圧力と風況のシミュレーション

本研究では、船体に作用する外乱成分として、風力を扱う。風によって船体に作用する surge, sway 方向の力および、yaw 方向のモーメントは次のように表される：

$$\left. \begin{array}{l} X_A = \frac{1}{2} \rho_A A_F U_A^2 C_{AX}(\psi_A) \\ Y_A = \frac{1}{2} \rho_A A_L U_A^2 C_{AY}(\psi_A) \\ N_A = \frac{1}{2} \rho_A A_L L_{OA} U_A^2 C_{AN}(\psi_A) \end{array} \right\} \quad (\text{A.14})$$

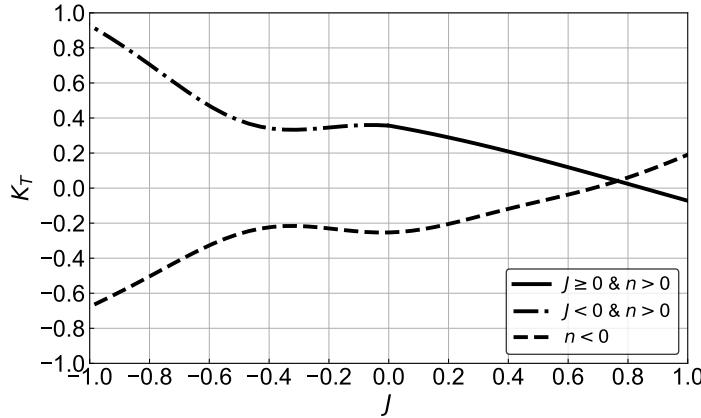


Figure A.2 Estimated K_T - J curves based on the estimate formula of MAU series propellers for the 1st quadrant and the database of B-series propellers for 2-4th quadrants.

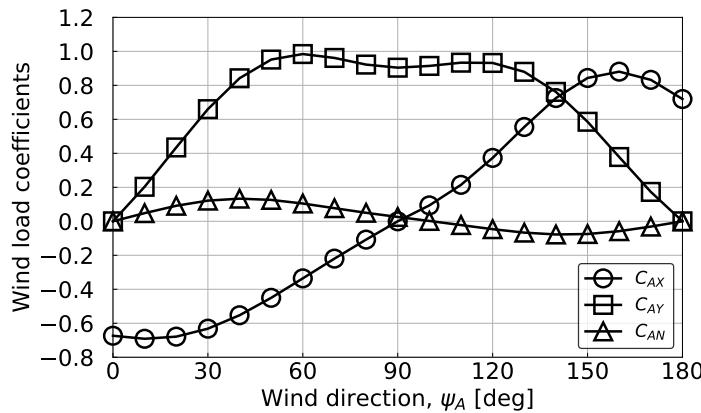


Figure A.3 Estimated wind load coefficients using method proposed by Kitamura et al.[9]

ここで、真風向風速と見掛けの風向風速は次のように表される：

$$\mathbf{U}_A = [u_A, v_A] = \begin{bmatrix} -U_w \cos \psi_w - u \cos \psi + v \sin \psi \\ -U_w \sin \psi_w - u \sin \psi - v \cos \psi \end{bmatrix}^T \quad (\text{A.15})$$

ここで、 U_w は真風速、 ψ_w は真風向、0 [deg] は真北からの風向を表す。 U_A は見掛けの風速、 ψ_A は見掛けの風向で 0 [deg] は船首方向から吹く風向きとする。真風向は、船の方位である真北から時計回りに度数で測定される。見かけの風向は、 $\psi_A = \tan^{-1}(v_A/u_A) - \psi_w$ として計算する。風圧力係数 C_{AX} , C_{AY} , C_{AN} は北村らにより提案された方法 [9, 10] で決定した。風圧力係数のカーブを Fig. 3.9 に示す。数値シミュレーションにおける風況は、風速・風向一定の場合と確率分布を用いたサンプリングの 2 つの設定で生成した。確率分布を用いたシミュレーションでは、真風

速は Weibull 分布、真風向は Gauss 分布に従って、以下のように生成した [11, 12]:

$$f_{U_W}(U_W | \bar{U}_W, a, \lambda) = \frac{a}{\lambda} \left(\frac{\bar{U}_W}{\lambda} \right)^{a-1} e^{-(U_W/\lambda)^a} \quad (\text{A.16})$$

$$f_{\psi_W}(\psi_W | \bar{\psi}_W, \sigma) = \frac{1}{\sqrt{2\pi}\sigma^2} \exp\left(-\frac{(\psi_W - \bar{\psi}_W)^2}{2\sigma^2}\right) \quad (\text{A.17})$$

ここで、 \bar{U}_W は平均真風速、 $\bar{\psi}_W$ は平均真風向である。Eq. (A.16) の形状パラメータ a は 2.0 とした。スケールパラメータ λ は、次のように計算される、

$$\lambda = \frac{2\bar{U}_W}{\sqrt{\pi}}. \quad (\text{A.18})$$

この式は、 $a = 2.0$ とする場合、ワイブル分布の期待値の式から得られる：

$$\bar{U}_W := \mathbb{E}[U_W] = \lambda \Gamma\left(\frac{3}{2}\right). \quad (\text{A.19})$$

ここで、 $\Gamma(x)$ はガンマ関数である。標準偏差 σ は、試験海域で測定された風向に従って 30 [deg] に設定した。Fig. A.4 は、シミュレーションと実船の自動着棧制御の結果の比較である。Fig. A.4 の上部はシミュレーションと実践試験中の風向風速の積み上げプロットであり、両者の分布は概ね一致しており、シミュレーションによる風況の再現ができていることが確認できる。また、航跡についても両者は概ね一致しており、風圧力の推定を含めた風外乱下の運動が再現できていることが確認された。

A.2 流体力微係数とその他のパラメータの同定

このモデルの流体力微係数やその他のパラメータは、海上試験での測定結果に基づき海上試験の測定値に基づいて同定した。改めて、神峰の主要目を Table A.1 に示す。プロペラ回転数 3.1 [rps] のアイドル状態で、舵角 20, 40 [deg] のプロペラ正転、逆転の回頭試験や、前方のみ $\pm 10, \pm 20$ [deg] の Z 試験など、さまざまな試験を実施した。次に、様々な推定式やデータベース [5, 6, 13, 14] を参照し、運動方程式の初期パラメータを設定した。ただし、これらの推定パラメータは異なる船型に基づくものであるため、神峰の実船試験の測定結果に基づいてパラメータの調整を実施した。この手順で調整したパラメータを Table A.2 に、その他本研究で使用したモデルのパラメータを Table A.3 に、別にプロペラモデルの係数を Table A.4 に示す。この際、潮流などの外乱の影響を含む回頭運動の計測値に対して定常状態のシミュレーション結果を比較することで、今回計測することのできなかった外乱の影響が小さくなるように考慮した。舵角とプロペラ回転数の変化について、実測値とシミュレーション結果の比較を Fig. A.5 に示す。シミュレーションにより、定常状態での回頭運動は十分に推定できることがわかる。また Z 試験に関しても、シミュレーション結果は測定結果とよく一致している。Fig. A.6 は、主機アイドリング回転数におけるプロペラ正転時の運動性能について、実測値と数値シミュレーションの結果を比較するための旋回試験の

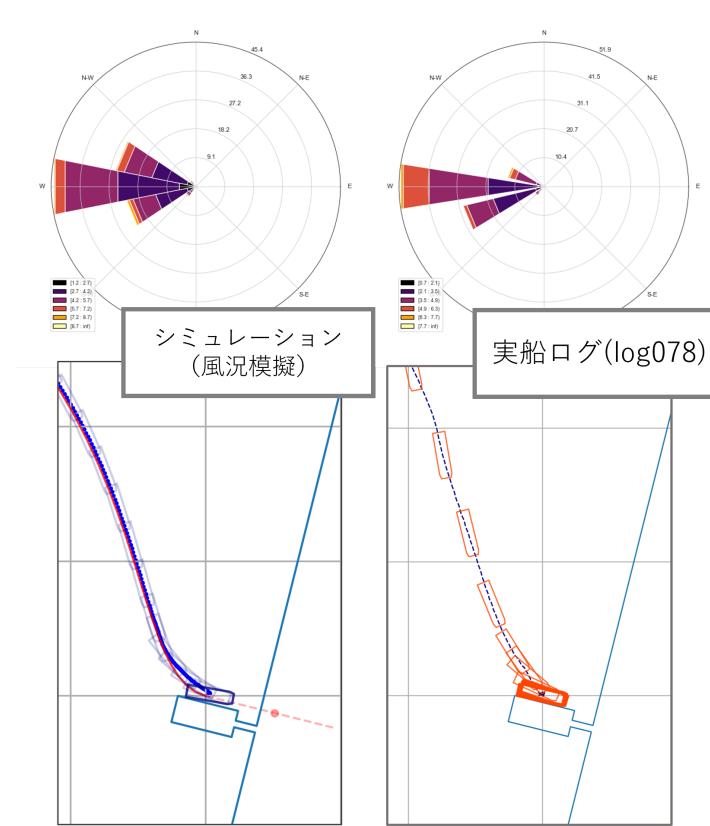


Figure A.4 Comparison of simulated wind and full-scale ship experiment during automatic berthing.

時系列を示したものである。Fig. A.6で実施した数値シミュレーションでは、実船での計測時の平均風速と風向に対応して風力を算出している。旋回航跡に違いはあるが、実船実験での潮流など測定できない影響をシミュレーションで可能な限り除外するため定常状態を再現したため、速度成分 u , v , r は実船の低速時の動きを再現できていることがわかる。

Table A.1 Principal particulars of the experimental ship "Shinpo" (Same table as the preceding section)

Subjects	Value
Ship length overall, L_{oa} [m]	16.5
Ship length between perpendiculars, L_{pp} [m]	14.9
Ship breadth (1.0WL), B [m]	4.38
Ship draft, d_m [m]	0.502
Trim, τ [m]	0.348
Diameter of propeller, D [m]	1.0
Propeller pitch @ 0.7R, P [m]	0.775
Side area of rudder, A_R [m^2]	0.780
Height of rudder, H_R [m]	1.05
Mass, m [kg]	21.53×10^3
Moment of inertia, I_{zG} [$kg \cdot m^2$]	3.044×10^2
Block coefficient, C_b	0.642
Longitudinal coordinate of center of gravity, x_G [m]	-1.260
Longitudinal projected area, A_F [m^2]	14.40
Lateral projected area, A_L [m^2]	41.52

Table A.2 Hydrodynamic force coefficients

Parameter	Ahead ($u \geq 0$)	Astern ($u < 0$)
Y'_v	-1.23	-0.5
Y'_r	0.106	0.0476
N'_v	-0.0791	-0.0949
N'_r	-0.134	-0.0614
C_D	0.185	0.0834
C_{rY}	1.41	6.36
C_{rN}	3.34	1.53
Parameter	Forward ($n \geq 0$)	Reverse ($n < 0$)
Y_P^*	-	-0.00924
x'_{PR}	-	-0.278
δ_0 [deg]	-2.47	5.0
x'_H	-0.364	0.0

Table A.3 Hydrodynamic coefficients and parameters for simulation

m'_x	0.0502	t_P ($n \geq 0$)	0.075	C_{tp0} ($J_p \leq -1$)	0.8294
m'_y	0.385	w_R	0.0	C_{tp1} ($J_p \leq -1$)	-0.0071
J'_z	0.00235	α_{rstl} [deg]	35.0	C_{tp0} ($-1 < J_p \leq J_{p0}$)	1.5320
X'_{0F}	-0.049	α_S [deg]	10.0	C_{tp1} ($-1 < J_p \leq J_{p0}$)	0.6954
X'_{0A}	-0.056	η	0.847	C_{tp0} ($J_{p0} < J_p$)	0.9246
X'_{vr}	0.055	C_{N0}	2.657	C_{tp1} ($J_{p0} < J_p$)	-0.8628
w_P	0.0	x'_R	-0.500	J_{p0}	-0.3898

Table A.4 Coefficients to representing force due to and rudder

Parameter	$n \geq 0$	$n < 0$
t_R	0.0897	0.642
a_H	0.169	-0.696
x_H/L_{PP}	-0.364	0.138
k_x	0.541	0.144
γ_R	0.452	1.0
l_R	-0.931	-0.5

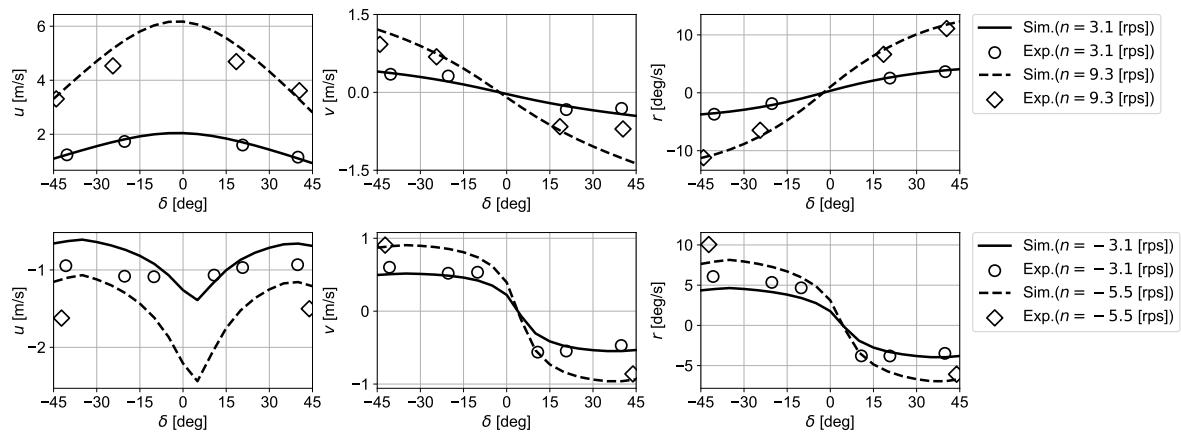


Figure A.5 Comparison on the full-scale ship experiment results and simulation results

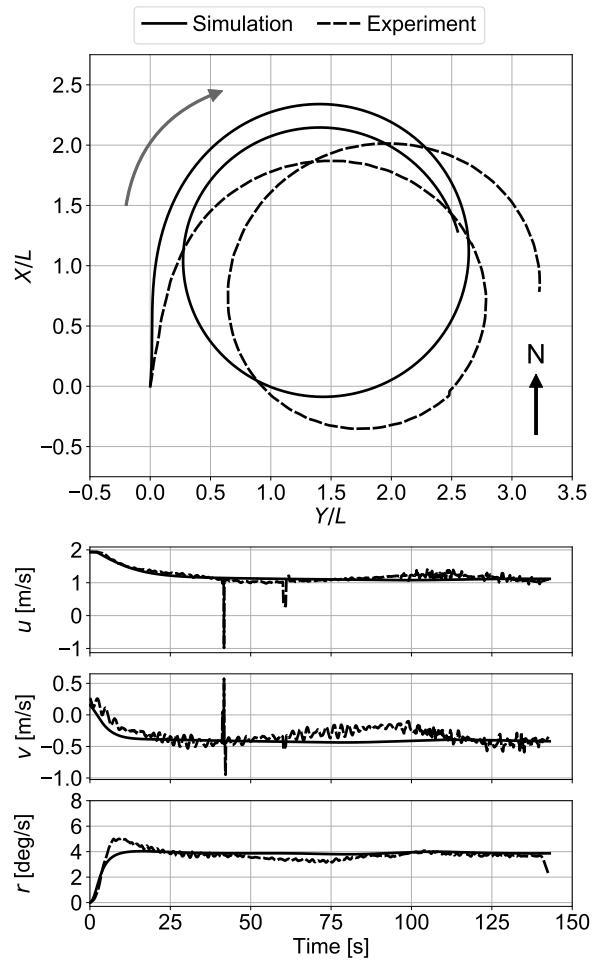


Figure A.6 Comparison of turning test results at idle speed between the full-scale ship experiment and the simulation. (Rudder angle is +45 [deg]. For full-scale ship experiment, the mean true wind speed is 2.08 [m/s], the mean true wind direction is 348.4 [deg].)

Appendix の参考文献

- [1] H. Yasukawa and Y. Yoshimura. Introduction of MMG standard method for ship maneuvering predictions. *Journal of Marine Science and Technology*, 20:37–52, 03 2014. doi: 10.1007/s00773-014-0293-y.
- [2] 元良誠三. 船体運動に対する附加質量及び附加慣性モーメントについて其の 1. 旋回に対する附加慣性モーメント. 造船協会論文集, 1959(105):83–92, 1959. doi: 10.2534/jjasnaoe1952.1959.83.
- [3] 元良誠三. 船体運動に対する附加質量および附加慣性モーメントについて. 造船協会論文集, 1960(106):a59–a62, 1960. doi: 10.2534/jjasnaoe1952.1960.a59.
- [4] 元良誠三. 船体運動に対する附加質量および附加慣性モーメントについてその 3 左右動に対する附加質量. 造船協会論文集, 1960(106):a63–a68, 1960. doi: 10.2534/jjasnaoe1952.1960.a63.
- [5] Yasuo Yoshimura, Ika Nakao, and Atsushi Ishibashi. Unified mathematical model for ocean and harbour manoeuvring. *Proceedings of MARSIM2009*, pages 116–124, 2009.
- [6] 北川泰士, 塚田吉昭, 宮崎英樹. ロペラ逆転時のプロペラ・舵の数学モデルに関する検討. 日本船舶海洋工学会講演会論文集, 20(0):117–120, 2015. ISSN 2424-1628. doi: 10.14856/conf.20.0_117.
- [7] 小瀬邦治. 操縦運動の数学モデルの具体化-船体プロペラ舵の相互干渉とその表現. 第3回操縦性シンポジウム, 0:27–80, 1981. URL <https://cir.nii.ac.jp/crid/1573668924357778304>.
- [8] 藤井斉. 自航模型船による舵特性の研究(1). 造船協会論文集, 1960(107):105–111, 1960. doi: 10.2534/jjasnaoe1952.1960.107_105.
- [9] Fumitoshi Kitamura, Michio Ueno, Toshifumi Fujiwara, and Naoto Sogihara. Estimation of above water structural parameters and wind loads on ships. *Ships and Offshore Structures*, 12(8):1100–1108, 2017.
- [10] 船舶風圧力簡易推定プログラムについて. 北村文俊, 上野道雄, 藤原敏文. 海上技術安全研究所報告, 3(9):207–213, 2009.
- [11] Adam Hugh Monahan. The probability distribution of sea surface wind speeds. part I: Theory and seawinds observations. *Journal of Climate*, 19(4):497–520, 2006.
- [12] J.V. Seguro and T.W. Lambert. Modern estimation of the parameters of the weibull wind speed distribution for wind energy analysis. *Journal of Wind Engineering and Industrial Aerodynamics*, 85(1):75 – 84, 2000. ISSN 0167-6105.
- [13] 芳村康男, 増本友美子. 中速商船・漁船船型の操縦流体力データベースと操縦運動の一推定法. 日本船舶海洋工学会論文集, 14:63–73, 2011. doi: 10.2534/jjasnaoe.14.63.
- [14] 塚田吉昭. 小型高速旅客船の性能向上に関する実験的研究. 船舶技術研究所報告 = *Papers of Ship Research Institute*, 28(5):p281–307, 09 1991. ISSN 0495775X.