



Title	着桟操船の自動化に向けた人工ニューラルネットワークを用いた船舶操縦モデルと軌道追従制御
Author(s)	脇田, 康希
Citation	大阪大学, 2025, 博士論文
Version Type	VoR
URL	<a href="https://doi.org/10.18910/101677">https://doi.org/10.18910/101677</a>
rights	
Note	

*The University of Osaka Institutional Knowledge Archive : OUKA*

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

The University of Osaka

# 博士学位論文

着棧操船の自動化に向けた  
人工ニューラルネットワークを用いた  
船舶操縦モデルと軌道追従制御

脇田 康希

2024年12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

## 要旨

着桟操船は、風や潮流などの外乱影響、岸壁や他船などの障害物の存在を考慮しつつ、複数の操船装置を適切に操作する必要があるため、船舶運航の中でも特に船員の負担が大きいオペレーションの一つである。そのため、本研究では、着桟操船の自動化に向けて、少ない計算資源で船舶の操縦運動をシミュレート可能な操縦モデルの推定手法と着桟操船シナリオを実行可能な軌道追従のための制御手法に関する研究に取り組んだ。本論文ではまず、人工ニューラルネットワーク（ANN）を用いた操縦モデルの推定手法を提案した。従来、ANNを用いた操縦モデルの推定には入出力関係を学習するために加速度計測が必要とされてきたが、実環境では計測ノイズや誤差が生じやすく、これが操縦モデルの精度低下を招く課題となっていた。そこで本研究では、操縦シミュレーションによる操縦運動軌道の予測誤差を最小化する最適化問題として定式化し、加速度の計測値を必要としないパラメータ推定手法を提案した。また、低速操縦運動の予測精度を向上させるため、従来のジグザグ操船や旋回操船のデータに加えて、ランダムにアクチュエータを操作した操船試験データを訓練データとして用いた。さらに、データ量が限定される状況でも推定精度を向上させるため、スライシングとジッタリングと呼ばれるデータ拡張手法の適用を提案した。これらの提案手法により、低速操縦運動を含む着桟操船軌道の予測精度が向上し、拘束模型試験や経験式に基づく MMG モデルと同等またはそれ以上の予測精度を持つ着桟操船のための操縦モデルが得られることを示した。本論文では次に、強化学習を用いた着桟操船軌道の追従ための制御方策の獲得手法を提案した。着桟操船では、障害物との距離が近く、衝突を避けながら多様な軌道パターンに対応する必要がある。着桟操船で遭遇する可能性のある多様な軌道パターンを学習させるため、ランダムにアクチュエータを操作された参照軌道を追従させる訓練手法を提案した。その結果、着桟操船軌道のような高速領域と低速領域の両方を含む軌道を追従可能な制御方策が得られた。また、制御方策に静的擬似障害物を考慮させ、報酬関数に衝突を引き起こす可能性のある追従誤差に追加のペナルティを反映させることで、着桟操船軌道の追従実験における衝突確率が減少することを示した。本提案手法は、訓練のための操縦シミュレーション環境を必要とするが、操縦運動データから ANN を用いて推定された操縦モデルを用いることが可能であることが示されている。そのため、本論文の提案手法により操縦運動データの計測により、着桟操船のための操縦モデルと軌道追従制御を獲得すること可能となった。本論文は以下の 6 章で構成されている。第 1 章は序論であり、研究背景、関連研究および研究目的について述べる。第 2 章では本論文で前提となる数学的表記法、船舶操縦運動に関する基礎知識、供試模型船の実験システム構成および検証用の拘束模型試験や経験式に基づく MMG モデルについて説明している。第 3 章では、順伝播型ニューラルネットワークおよび回帰型ニューラルネットワークに基づく操縦モデル構造、および、加速度の計測値を必要とする回帰問題ベースのパラメータ推定手法と提案する軌道推定ベースのパラメータ推定手法についてそれぞれ説明し、自由航走模型試験データを用いた提案手法の検証実験結果を示した。第 4 章では、スライシングおよびジッタリングと呼ばれるデータ拡張手法と、データ拡張手法の第 3 章で述べた軌道推定ベースのパラメータ推定手法への適用手法について説明し、自由航走模型試験データを用いてデータ拡張手法の操縦モデルの推定精度に対する影響を示した。第 5 章では、強化学習の概要を説明した後、操縦シミュレーション環境構成、報酬関数設計、参照軌道の生成手法、静的擬似障害物の生成手法、を含む軌道追従制御方策の獲得手法について、シミュレーションおよび模型船における着桟操船軌道の追従実験や 4 コーナー DP 実験の結果を示した。第 6 章では、以上の内容を総括し、今後の課題と展望を述べる。

## 謝辞

本学位論文は、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での研究成果をまとめたものです。本研究は、日本学術振興会（JSPS）科学研究費助成事業（JSPS KAKENHI Grant #23KJ1432）の支援を受けて実施されました。また、本研究の遂行にあたり、多くの方々からご指導・ご支援を賜りました。ここに深く感謝の意を表します。

まず初めに、指導教員である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の牧敦生教授に、心より感謝申し上げます。牧教授には、学部4年生の頃から長きにわたり、研究の方向性の指導、課題解決の助言、論文執筆のご指導など、あらゆる方面から研究に関する多大なるご指導をいただきました。また、情報科学や航空宇宙、生命科学分野などさまざまな異分野の研究者との議論の場を設けていただきなど、多くの貴重な機会をいただきました。博士課程への進学を決断し、本学位論文を執筆するに至ったのも、牧先生のご指導とお力添えがあってのことと深く感謝しております。改めて、心より御礼申し上げます。

また、学位論文の副査をご担当いただきました、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の箕浦宗彦教授、酒井政宏准教授、ならびに筑波大学システム情報系の秋本洋平准教授に、心より感謝申し上げます。箕浦教授には、学位審査における的確なご指摘とアドバイスを賜りました。酒井准教授には、研究に関する日々の議論に加え、国際学会での研究発表や学生生活における様々なご助言を賜りました。秋本准教授には、機械学習に関する理論や応用方法について、幅広くかつ多岐にわたるご指導を賜りました。さらに、論文執筆に際しては、細部に至るまで鋭く的確なご指摘をいただき、夜遅くまでのミーティングにもお付き合いいただきながら、根気強くご助言をいただきました。この場を借りて、改めて深く御礼申し上げます。

研究活動を支えてくださった秘書の岡田祐子様、ならびに、大阪大学船舶知能化領域の先輩、後輩、同輩の皆様にも心より感謝申し上げます。皆様との議論や交流は、私の研究を深める大きな原動力となりました。また、私の大学院進学を快く応援し、これまでの学生生活における多大なる支えとなってくれた両親に深く感謝いたします。最後に、研究を支え、励ましてくださったすべての方々に改めて感謝申し上げます。皆様のご支援とご助力がなければ、本学位論文を完成させることはできませんでした。この場を借りて、心より御礼申し上げます。

# 目次

要旨	i
謝辞	ii
<b>第 1 章 序章</b>	1
1.1 研究背景 . . . . .	1
1.1.1 自動運航船 . . . . .	1
1.1.2 着桟操船および港内操船の自動化 . . . . .	2
1.2 関連研究 . . . . .	3
1.2.1 操縦モデルの推定手法 . . . . .	3
1.2.2 操船制御アルゴリズム . . . . .	6
1.3 研究目的 . . . . .	8
1.4 自著論文(学術雑誌掲載論文および学会論文) . . . . .	10
<b>第 2 章 事前知識</b>	12
2.1 表記法 . . . . .	12
2.2 平水中における 3 自由度の操縦運動 . . . . .	12
2.2.1 座標系 . . . . .	12
2.2.2 操縦運動方程式 . . . . .	14
2.3 供試船 1: 一軸一舵を搭載した模型船 . . . . .	15
2.3.1 自由航走試験のため計測および操船システム . . . . .	15
2.3.2 操縦シミュレーションのための MMG モデル . . . . .	16
2.4 供試船 2: VecTwin 舵システムを搭載した模型船 . . . . .	19
2.4.1 自由航走試験のため計測および操船システム . . . . .	19
2.4.2 操縦シミュレーションのための MMG モデル . . . . .	20
2.5 実験池: 犬飼池 . . . . .	21
<b>第 3 章 人工ニューラルネットワークを用いた操縦モデルの推定手法</b>	22
3.1 操縦モデルの構造 . . . . .	22
3.1.1 順伝播型ニューラルネットワーク . . . . .	22
3.1.2 回帰型ニューラルネットワーク . . . . .	24
3.1.3 入出力変数の前後処理 . . . . .	26
3.1.4 操縦モデルの表記方法 . . . . .	27

3.2	操縦モデルの推定手法 . . . . .	27
3.2.1	回帰分析ベースの手法 . . . . .	27
3.2.2	軌道推定ベースの手法 . . . . .	28
3.2.3	ANN パラメータの最適化手法 . . . . .	30
3.3	模型試験データを用いた検証実験 . . . . .	31
3.3.1	操縦運動データ集合 . . . . .	31
3.3.2	実験内容 . . . . .	32
3.3.3	実験結果 . . . . .	35
3.4	議論 . . . . .	38
3.5	結言 . . . . .	40
第 4 章	人工ニューラルネットワーク用いた操縦モデル推定のためのデータ拡張手法	41
4.1	データ拡張手法 . . . . .	41
4.1.1	部分列 . . . . .	41
4.1.2	スライシング . . . . .	42
4.1.3	ジッタリング . . . . .	43
4.2	模型試験データを用いた検証実験 . . . . .	44
4.2.1	操縦運動データ集合 . . . . .	44
4.2.2	実験内容 . . . . .	47
4.2.3	実験結果 . . . . .	50
4.3	議論 . . . . .	53
4.4	結言 . . . . .	54
第 5 章	強化学習を用いた軌道追従制御方策の獲得手法	56
5.1	強化学習の概要 . . . . .	56
5.2	軌道追従制御方策の獲得手法 . . . . .	57
5.2.1	操縦運動のシミュレーション環境 . . . . .	57
5.2.2	軌道追従制御問題 . . . . .	59
5.2.3	静的障害物の考慮 . . . . .	64
5.3	未知の参照軌道および静的障害物への適用 . . . . .	68
5.3.1	参照状態の選択方法 . . . . .	68
5.3.2	擬似障害物の生成 . . . . .	68
5.4	MMG モデルを用いた検証実験 . . . . .	69
5.4.1	追従制御方策の訓練結果 . . . . .	69
5.4.2	着棧操船の軌道計画 . . . . .	70
5.4.3	着棧操船軌道の追従結果 . . . . .	71
5.5	操縦運動データを用いた実験 . . . . .	77
5.5.1	操縦モデルの推定結果 . . . . .	78
5.5.2	追従制御方策の訓練結果 . . . . .	79
5.5.3	参照軌道の追従結果 . . . . .	80

5.6	議論 . . . . .	85
5.7	結言 . . . . .	86
第 6 章	総括	87
参考文献		89

# 第1章

## 序章

本章では、本論文の導入として、まず研究背景である自動運航船の定義やその目的について概説する。その後、自動運航船における課題の一つである着棧操船および港内操船の自動化に関する問題に触れ、本研究の動機を示す。また、関連する文献のレビューを通じて、抽出された研究課題と本研究の目的について論じる。最後に、本論文に含まれる自著論文の一覧を提示する。

### 1.1 研究背景

#### 1.1.1 自動運航船

近年、海運業界では自動運航船 (Maritime Autonomous Surface Ships: MASS) に関する研究開発が活発化している。MASSは、国際海事機関 (International Maritime Organization : IMO) によって“人間の介入が全く、もしくは、ほとんどなしに運航上のさまざまなオペレーションを実行できる船舶”と定義される船舶の総称である。ただし、MASSには、船舶運航のための作業や意思決定が完全に自動化・自律化された船舶のみだけではなく、段階的に自動化・自律化された船舶も含まれる。その自動化レベルは以下の4段階に分類されている。

- レベル 1: 船内作業の一部自動化と操船における意思決定支援。船員の主な役割は船内システムの操作および管理であり、船内作業は一部自動化されているが必要に応じて船員が代替可能な状態で運行される。
- レベル 2: 船員が乗船している遠隔操作船。船舶は別の場所から制御および操作されるが、船員は乗船しており、必要に応じて作業を代替する。
- レベル 3: 船員が乗船していない遠隔操作船。船舶は別の場所から制御および操作され、船員は乗船していない状態で運行される。
- レベル 4: 完全自律型船。船舶のオペレーティングシステムがすべての意思決定を行い、操船が行われる。

このような自動化・自律化レベルの段階的な向上を目指して、欧州や日本を中心に、多くの企業や研究機関が実用化に向けた研究開発に取り組んでいる。例えば、日本財団が主導する“MEGURI2040 Fully Autonomous Ship Program”において、複数のコンソーシアムが完全自律船の開発が実施されている。このプロジェクトでは、貨物船や観光船、フェリーなど多様な船種を対象とした実証実験が行われ、特に自動着棧技術の実用化

に向けた研究開発が進められている。また、技術的な研究開発に加えて、IMO や各国の船級協会では MASS に課される規則やガイドラインに関する議論が進められている。

MASS の実用化は以下のような利点を海運業界にもたらすと考えられている。

- 人件費の削減: 船員の労働負担を軽減することで、乗組員に要する経費を削減できる。
- 安全性の向上: 疲労による人為的ミスを軽減し、危険な環境での作業を削減することにより、海上事故を減らすことができる。
- 運航の効率化: 航路や船速の最適化により、運航効率や燃費効率が向上させることができる。

現在、日本では高齢化が進行しており、内航海運においても人材不足が深刻化しつつある。船員には高度な知識とスキルが要求されるが、それらの習得には長い時間が必要であり、ベテラン層のリタイアに伴い人材不足はさらに深刻になると予想される。この点からも、船員の労働負担を軽減する MASS の実用化は喫緊の課題である。

着桟操船は船舶運航の中で最も船員が負担を感じるオペレーションの一つである。船舶の衝突事故は致命的な人的および経済的損害をもたらす可能性が高いが、着桟操船では外洋航行と比較して他船や障害物との距離が非常に近い。そのため、着桟操船は最新の注意と慎重な操作が求められるセーフティクリティカルなタスクである。しかし、着桟操船は技術的に難易度の高い作業であるため、熟練した船員であってもミスを犯す可能性は少なくない。この技術的な難しさの主な要因を以下に示す:

- 操船の複雑さ: 着桟操船では、前進速度が比較的大きく安定した標準的な操縦運動に比べて、大きな斜航角を取る低速操縦運動が頻繁に発生する。特に、着岸操船ではクラビング(横移動)、旋回、後進などの特殊な操船操作が要求される場合がある。着桟操船では、そのような操船を実現するために、複数のアクチュエータを適切に操作することが求められる。
- 外乱影響の強さ: 低速状態では、操舵力や船体流体力の低下により、風のような環境外乱による影響が比較的大きくなる。
- 障害物との近さ: 障害物との衝突回避だけでなく、適切な距離を保って操船される必要があり、操船の自由度が制限される。

従って、着桟操船の自動化は船員の負担を軽減する重要な課題であり、本研究ではこの課題に取り組んだ。

### 1.1.2 着桟操船および港内操船の自動化

着桟操船に関する用語として港内操船が存在するが、本論文ではまず、着桟操船および港内操船の定義を行う。船舶の着桟操船は船舶や港湾の種類によって様々な方法が用いられるが、本研究では、タグボートの支援を受けない大型船舶を想定し、宮内 [1] の定義を基に着桟操船および港内操船に関わる一連の操船を以下のように定義する。

- 入港操船: これは水道航路を通じて外洋航海および港湾を移動する操船を指す。この操船では、規則で決められた水路や制限船速を守り、他船との衝突を避けながら、目的地へ向かって航行することが求められる。
- アプローチ操船: これは着桟のために目的の接岸位置へ移動する操船を指す。この操船では、船速を適切に遅減させながら、周囲の他船や障害物との衝突を避け、最終的に接岸位置付近で船舶を停止もしく

は十分に減速させることが求められる。

- 接岸操船・着岸操船: これは、アプローチ操船の完了後、係留索で船舶が固定可能な位置まで船舶を移動させる操船を指す。大型船舶では、船首方位を一定に保ったまま平行横移動するように操船されることが多い。
- 着桟操船: これはアプローチ操船および接岸操船・着岸操船を含む港湾内の船舶を桟橋や岸壁へ移動させる一連の操船を指す。
- 港内操船: これは上述の四つの操船を全て含む港湾内で行われるすべての操船を指す。

したがって、本論文では、アクチュエータ(舵、プロペラ、スラスターなど)を適切に操作して船舶を港湾のある地点から目標地点まで安全かつ自動的に導くことを“港内操船の自動化”と定義し、特に、目標地点が接岸位置の場合は“着桟操船の自動化”と定義する。つまり、この“着桟操船および港内操船の自動化”が本論文の目的となる。

着桟操船および港内操船の自動化のための操船システム開発ではさまざまな要素技術が必要される。ここでは、必要とされる要素技術のいくつかの例を以下に示す:

- 操船制御アルゴリズム: 与えられるシナリオを実現するようにアクチュエータの動作を決定する技術。アクチュエータの動作は、船舶の操縦性を考慮した上で、自船の状態（位置、方位、速度、舵角、回転数など）や外部環境の状態（海岸地形、水深、水路位置、風況、海面状況、他船位置など）に応じて適切に判断される必要がある。
- 状況認識技術: 自船の状態および外部環境の状態に関する情報を観測するセンシングおよびセンシングおよび状態推定技術。従来のセンサに加え、船員の目視から得られる情報を自動的に観測する必要があり、また、海底危険物、航路標識などの海図情報や港則法のような航行規制情報も併せた状況の把握が必要である。
- 操縦モデル（操船シミュレータ）: 少ない計算資源で船舶の操縦運動をシミュレート可能な船舶のためのシステムベース数理モデル。制御アルゴリズムの設計および評価のために使用することが可能で、実船を用いた検証の回数を減らす可能性があるため、実用的なツールとして広く利用されている。

本研究では、このうち、着桟操船および港内操船の自動化に向けた“操縦モデル”および“操船制御アルゴリズム”的獲得技術に関する研究に取り組んだ。

## 1.2 関連研究

### 1.2.1 操縦モデルの推定手法

まず、操縦モデルの推定に関する関連研究について述べる。操縦モデルは船舶の操縦運動をシミュレートするためのシステムベースの数理モデルであり、運動学変数に関する常微分方程式や状態遷移関数を通して操縦運動を表現する。操縦モデルは、数値流体力学（Computational Fluid Dynamics: CFD）に比べて少ない計算量でシミュレーションが可能であるため、造船設計における操縦性能の予測や、操船制御アルゴリズムの設計、操船シミュレータの開発など、さまざまな用途で活用されている。以下では、いくつかのアプローチ手法に基づいた関連研究について述べる。

### (1) 水槽試験に基づくアプローチ

多くの操縦モデルは流体力学に基づいて構築されている。代表的な操縦モデルとして、日本試験水槽委員会 (Japan Towing Tank Conference: JTTC) の研究グループによって提唱された Maneuvering Modeling Group (MMG) モデルが挙げられる。MMG モデルは船体、プロペラ、舵、その他のアクチュエータによって生じる流体力を表現するサブモデルから構成される数理モデルである [2]。このモデルはモジュール性を持ち、造船過程において部分的な設計変更が発生しても、特定のサブモデルの変更のみで対応可能であるという利点を持つ。通常、MMG モデルは標準的な操船に焦点が当てられているが [3]、低速操縦運動に対応した MMG モデルに関する研究も多数行われている [4-8]。

操縦モデルに含まれる流体力の大部分は、縮尺模型船を用いた拘束模型試験 (Captive Model Tests) で測定可能である。この試験方法には、平面運動機構 (Planar Motion Mechanism: PMM) 試験 [9] や Circular motion tests (CMT) [10] がよく知られている。これらの試験で計測した力を基に、操縦モデルの構造やパラメータを決定することができる。さらに、いくつかのパラメータは経験式によって決定可能であり、例えば付加質量や附加慣性モーメントの決定には元良のチャート [11] が、船体流体力の係数には井上の式 [12]、貴島の式 [13]、芳村の式 [8] などが使用される。また、舵直圧力勾配には藤井の式 [14]、船体の風圧力には藤原の式 [15] が利用可能である。

ただし、拘束模型試験を実施するためには、曳航水槽や操縦水槽といった特別な試験設備や、それらを適切に運用する専門的知識が必要となる。対象船舶によっては、経験式や過去の拘束模型試験データの活用が可能な場合もあるが、流体力を精密に表現しようとするにつれ、モデルの複雑さが増し、必ずしもすべてのパラメータが決定できるわけではない。そのため、試験水槽を有する造船所や研究機関を介さないと、拘束模型試験に基づく操縦モデルの獲得は難しい。また、フルスケールの船舶で計測可能な力はほとんどないため、模型スケールでの試験結果を適切にスケーリングする必要があるが、スケール影響の問題に直面する可能性がある。

### (2) システム同定に基づくアプローチ

操縦モデリングの異なるアプローチの一つとして、システム同定 (System Identification: SI) がある。SI を用いた操縦モデルの獲得手法は長年にわたり研究が行われてきた [16-24]。SI は、力の計測が必ずしも必要なく、運動学的変数と制御入力の時系列データからモデル化が可能であるため、既存の船舶であれば、フルスケール船であっても、操縦モデルを直接推定できるという利点を持つ。SI では、データに対してモデルがどの程度実際の応答結果を表現しているかを評価する閾数が定義され、実際の応答結果を十分に説明できるモデルが作成される。従来の SI に基づく操縦モデリング研究は、以下の 4 つの観点に基づいて部分的に分類できる。

#### 操縦モデルがどのように定義されるか

操縦モデルは、大きく離散システムか連続システムの 2 種類に分類される。離散システムでは、現在の状態変数と入力変数から次の時刻の状態変数を予測する状態遷移モデル [25] や、現在および過去の状態変数や入力変数から次の時刻の状態変数を予測する非線形自己回帰外因性 (Nonlinear Autoregressive Exogenous: NARX) モデル [26-28] が用いられることが多い。これらのモデルには人工ニューラルネットワーク (Artificial Neural Network: ANN) [25, 28]、サポートベクターマシン (Support Vector Machine: SVM) [29]、ガウス過程 (Gaussian Processes: GP) [27] などの機械学習サロゲートモデルが頻繁に利用される。一方、連続システムでは、常微分方程式によって状態変数とその時間微分の関係をモデル化する方法が採られ、MMG モデル [22, 23] や Abkowitz モデル [17, 24, 30] が代表的である。

連続システムでも SVM [31, 32] や ANN [28] といった機械学習サロゲートモデルで近似する例がある。評価関数がどのように定義されるか

SI の評価関数は、モデルの入出力関係の一致度や一連の操縦運動の応答結果の一致度に基づく指標が使用される。前者の入出力関係に基づく手法は回帰分析に基づき、出力の予測値と観測値の誤差が最適化基準として用いられるが、連続システムモデルの場合は加速度や力の計測が必要である。後者の応答結果に基づく手法では、状態遷移関数の繰り返し計算や常微分方程式の数値解法を用い、一連の操縦運動の応答を求める。この場合、実際の応答データとの二乗誤差 [22]、ハウドルフ距離 [20]、観測ノイズに対する負の対数尤度 [33] などが最適基準として利用される。この手法は比較的容易に計測可能な位置や速度のみで評価基準を設定できるため、必ずしも加速度か力に関する計測値を必要としない。

パラメータがどのように最適化されるか

SI における最適化手法は、操縦モデルやその用途に応じて選択される。例えば、パラメトリックモデルには拡張カルマンフィルタ (Extended Kalman Filter: EKF) [16–19] が古くから用いられているが、パラメータ数が増加するとキャンセレーションの影響を受けやすくなることが報告されている [17]。また、サポートベクター回帰 (Support Vector Regression: SVR) [29, 31, 34, 35] のような回帰分析多くの研究で採用されており、多重共線性に対処しやすい利点がある。ただし、オンライン同定が不要な場合、EKF などのオンライン手法よりもすべての測定データにアクセス可能なオフライン同定の方が効果的とされる [20]。さらに、遺伝的アルゴリズム (Genetic Algorithm: GA) [20, 23] や共分散行列適応進化戦略 (Covariance Matrix Adaptation Evolution Strategy: CMA-ES) [22, 24, 36] を用いた進化戦略ベースのオフライン SI 手法も提案され、その有効性が示されている。

どのようなデータが与えられるか

SI には、実船や模型船による自由航走試験から得られる操縦運動データが用いられ、データの質や分布が SI の結果に大きな影響を与える。データ分布は、自由航走試験の操縦運動内容に応じて変化するが、多くの研究でジグザグ試験や旋回試験が使用される。また、ランダム操舵による操縦運動を使用する研究 [19] や、フィッシャー情報行列に基づきパラメータ同定誤差の分散を最小化する D-最適化による操舵運動を利用する研究 [37] もある。

また、いくつかの文献 [32, 38] では、従来の SI に基づく操縦モデリング研究を以下のように分類している。

- パラメトリック SI: 対象システムに関する先駆的な情報に基づく SI。先駆的な情報はしばしば数理モデルに組み込まれており、例えば MMG モデル [22, 23, 36, 39] や Abkowitz モデル [17, 24, 29, 30] が挙げられる。この手法は、先駆的な情報を基に簡潔かつ効率的にシステムを表現できる場合には、物理的仮定の恩恵により比較的少ないデータでパラメータを決定する可能性がある。しかし、システムの表現が不適切であると精度の低下を招く。
- ノンパラメトリック SI: 対象システムの先駆的な情報を用いない SI。ANN やカーネル関数を用いた SVM や GP といった機械学習ベースのサロゲートモデルが利用される。この手法では、多くのデータが要求される可能性が高いが、モデル構造を先駆的に決定する必要がなく、あらゆるアクチュエータ構成に適用可能であるため、流体力学の知識が浅いユーザにも実用的な手法といえる。

いずれの SI 手法にも利点と欠点があり、一般的にどちらが優れているかについての明確な結論は得られていない。よって、事前知識や与えられるデータに応じて選択されるべきと考えられるが、本研究ではさまざまなアクチュエータ構成に適用可能なノンパラメトリック SI 手法に焦点を当てる。

操縦モデルの推定のためのノンパラメトリック SI 手法において、ANN は最も代表的な手法の一つである。ANN は強力な関数近似能力を有し [40, 41]、操縦モデルの非線形性を十分に表現できることが多い研究 [25, 28, 38, 42–44] で示されている。Moreira ら [25, 42] は、ANN で表された操縦モデルを再帰的に用いる操船シミュレーションモデルを提案し、シミュレーションデータおよび実スケールデータによる検証を実施した。Rajesh ら [43] は、3 自由度の操縦方程式の非線形項を ANN で表現し、大型タンカーの非線形操縦モデルを推定した。Zhang ら [44] は Chebyshev 直交基底関数を持つ ANN を用いて非線形流体力を表現する関数を推定した。He ら [38] は、ベイズ最適化を用いたハイパーパラメータ探索による ANN モデルの自動設計手法を提案している。また、履歴影響を考慮するため、回帰型ニューラルネットワーク (Recurrent Neural Network: RNN) [26, 28, 45] や Long Short Term Memory (LSTM) [46, 47] を適用した操縦モデル推定に関する研究も行われている。

カーネル関数に基づく SVM [31, 37] や GP [27, 48, 49] も、操縦モデルの推定において代表的なノンパラメトリック SI 手法である。Luo ら [31] は実船試験のデータを用いて双胴船の特定の旋回円操縦運動を推定する SVM ベースの操縦モデルを提案した。Wang ら [37] は、 $\nu$ -SVR に基づくノンパラメトリックモデリング手法を提案し、KVLCC2 型タンカーの模型実験データで検証を行った。Ramirez ら [48] は、多出力 GP を用いてコンテナ船の 4 自由度操縦モデルを推定する手法を提案した。Xue ら [27] は、操縦モデルの推定精度の向上を目的にノイズ入力を考慮した GP 回帰手法を提案した。これらの手法はカーネルトリックを活用し、低い計算コストで操縦モデルの非線形パターンを捉えることが可能であり、構造リスク最小化を通じてオーバーフィッティングを緩和する [37]。また、モデルの適応向上を目指したオンライン学習に関する研究も進められている [49]。

港内操船には、比較的大きな前進速度による安定した操縦運動に加え、大きな斜航角の運動やアクチュエータの使用頻度が増加する低速操縦運動も含まれる。そのため、港内操船に適した操縦モデルは、状態変数の範囲や行動変数の次元が増大する傾向にある。ノンパラメトリック SI 手法の中でも ANN は高次元かつ大規模なデータに対応できる能力を持っており、港内操船のための操縦モデル推定に適していると考えられる。しかし、これまでの ANN を用いた操縦モデルのノンパラメトリック SI 手法に関する研究は、主に標準的な操縦運動に限定されており、港内操船向け操縦モデル推定における ANN の有効性を示す研究例は見当たらない。そこで、本研究では、ANN を用いた着桟操船および港内操船に適した操縦モデルの推定手法を検討している。

### 1.2.2 操船制御アルゴリズム

船舶の操船制御アルゴリズムに関する関連研究について述べる。与えられた計画を実行する操船制御アルゴリズムは、輸送のみならず、測量や地図作成など多様な用途に対応する重要な技術として注目されており、離着桟操船 [50–54]、衝突回避 [55, 56]、定点保持 [57, 58] などのさまざまなシナリオに適用されている。本研究の目的は着桟操船および港内操船の自動化であるため、ここでは、着桟操船の自動化に関する関連研究を示した後、本研究で取り組む軌道追従制御に関する研究について述べる。

#### (1) 着桟操船制御

近年、MASS の研究開発の進行に伴って着桟操船の自動化に関する研究が活発化しているが [59]、日本では、小山ら [60] による最適レギュレーター問題の適用を契機に、早期から高度自動運航システムの開発を目指した着桟操船の研究が多く報告されている [61–65]。例えば、小瀬ら [61] は二つの操船パターンの参照軌道を決定し、フィードバック制御及びフィードフォワード制御により着桟操船制御を試みている。ここではセーフティ

ファーストの原則から、計画において、岸壁から少し離れた仮のゴールを目標点とし、また、操作量余裕を持たせることで安全性の向上を試みている。さらに、小山ら [62] は着桟制御問題をアプローチ操船と着岸操船へ分割して取り組んでいる。特に着岸操船では、線形化した操縦モデルに対し最適レギュレーター問題として定式化しており、加えて、操船者の経験による着桟パターンのルールを活用し、評価関数の定義を行なっている。

着桟操船の自動化には、モデル予測制御(Model Predictive Control: MPC)を用いた最適制御問題(Optimal Control Problem: OCP)の解法 [66–68] や、適切な制御入力を ANN に学習させる手法 [53, 69–72] が提案されているが、多くの研究では自律航法問題を計画(Planning)と制御(Control)に分離している。例えば、正司ら [50, 73–75] は着桟制御問題を非線形の 2 点境界値問題として定式化し、Sequential Conjugate Gradient Restoration Method [76] により数値的に最適解を求めた。その後、最適制御解をフィードフォワード情報、最適軌道の状態誤差をフィードバック情報として用いた追従制御により着桟操船の自動化を行なった。Ahmed ら [53] は Imaginary line と呼ばれる基準線を導入し、PD 制御により基準線に沿うことでアプローチ操船の自動制御を行っている。また、Martinsen ら [77] は OCP に基づく着桟航路計画手法を提案し、Bitar ら [51] や Martinsen ら [52] は DPS を用いて計画軌道の追従することで着桟操船の自動化を実現している。澤田ら [21] は PD 制御による Pure Pursuit 法と 3 段階クラッチ制御でベジェ曲線を追従し、着桟操船の自動化を実現している。着桟操船の自動化において、計画と制御の分割はアルゴリズムの簡素化とリアルタイム性能の向上に効果があるため、多くの研究で採用されている。

着桟操船の自動化のための航行計画では、船舶を目標地点へどのように到達させるかを決定する必要があり、計画航路の定義方法は大きく以下に分類できる。

- 経路計画(Path planning)：船舶が通過すべき位置と方位の集合を定義する方法で、Ahmed ら [53] や澤田ら [21] の研究で用いられている。直線や曲線で定義されるため、計算量が少なくシンプルに実行可能だが、動的制約を考慮していないため実現可能な計画とならない場合がある。
- 軌道計画(Trajectory planning)：位置や方位に加えて時間的情報を含む計画手法であり、正司ら [50] の研究では、操縦モデルによる動的制約を考慮した実現可能な軌道が得られている。また、空間制約も考慮され、Martinsen ら [77] は安全マージンをポリゴンで定義し、空間制約を OCP に組み込んでいる。さらに、宮内ら [78] は牧ら [79, 80] の最適軌道計画手法を拡張し、Ship Domain を導入して障害物からの安全距離を確保している。ただし、非線形計画法を解く必要があり、計算コストが高くなる。

着桟操船では、外洋航海から入港とアプローチ操船を経て着岸操船に至るまで、船舶の減速能力を踏まえ船速を段階的に低下させる必要がある [81]。本論文では、船舶の操縦性やアクチュエータの限界が考慮された実現可能な軌道計画が与えられると仮定し、軌道追従制御(Trajectory Tracking Control)に関する研究に取り組んでいる。

## (2) 軌道追従制御

軌道追従制御は、船舶を与えられた軌道計画に従わせるためアクチュエータの動作を指示する役割を担っており、高い追従精度を保持するためには、操縦モデルの非線形性、モデル化誤差、環境外乱に応じた適切な判断が必要である。軌道追従制御に関する研究も多く進められており、非線形 MPC [47, 82, 83] やバックステッピング制御 [84–86] を用いたアプローチが検討されている。また、波、風、潮流といった外乱下での姿勢維持や修正を目的とした低速操船・定点保持に特化した制御システムである Dynamic Positioning System (DPS) [57, 58] を活用し、着桟操船シナリオにおける軌道追従制御を実施する研究も存在する [51, 52, 87]。

強化学習 (Reinforcement Learning: RL) [88] は、環境の不確実性を考慮できる特性から、船舶の運動制御において注目を集めている [55, 89–93]. RL は、環境制約および不確実性の下で、与えられた報酬関数に基づき最適な制御方針を探索するために利用される。また、RL と ANN を組み合わせることで、性能の大幅な向上が得られることが知られている [94]. 例えば、Martinsen らは、RL を用いて直線経路 [95] および曲線経路 [90] に対する追従誤差を最小化する制御器を構築し、その後、軌道追従のための制御器について検討している [91, 93].

軌道追従制御は、動的制約や空間制約、環境外乱などの不確実性などを考慮した上で、リアルタイム性を保ったまま、実行され続けなければならない、リアルタイムでの反復試行を伴う最適化計算は好ましくない。RL は軌道追従制御のための ANN を訓練することが可能であり、訓練された ANN は繰り返し計算処理をすることなく最適な制御入力を決定することができる。そのため、本研究では軌道追従制御のための ANN の RL 訓練手法について焦点を当てる。

### 1.3 研究目的

本研究では、ANN を用いた着桟操船および港内操船に適した操縦モデルの推定手法と、船舶の軌道追従制御を実現するための ANN の RL 訓練手法について検討する。本節では、それぞれの研究テーマに関連する残された課題と本研究の目的を整理する。

#### (1) ANN を用いた着桟操船および港内操船のための操縦モデルの推定手法

1.2.1 節で述べたように、ANN を用いた操縦モデルのノンパラメトリック SI の有効性は多くの研究で示されているが、低速操縦運動のための操縦モデル推定における ANN の有効性が確認された研究は知る限り存在しない。そのため、本研究の目的の一つは ANN を用いた港内操船のための操縦モデル推定手法を提案することである。最終的な目標は実船の操船モデルの予測であるが、ここでは、研究の第一段階として模型試験データを用いた検証実験を実施する。以下に、本研究で取り組む具体的な課題とその貢献を示す。

- これまで操縦モデルの入出力関係を学習するために加速度の計測が必要とされてきたが、実環境における加速度データはノイズや計測誤差の影響を受けやすい。そのため、本研究では ANN および RNN を用いた操縦モデルの推定手法として、操縦運動軌道の予測誤差を最小化する加速度を必要としない訓練手法を導入することを提案する。
- 多くの SI に関する既存研究では、ジグザグ操船や旋回操船のデータを用いて操縦モデル推定が行われているが、これらのデータは必ずしも港内操船や着桟操船の操縦モデル推定に適しているとは言えない。ジグザグや旋回操船は比較的大きな船速での標準的な操縦運動を計測するものであり、クラビング、旋回、後進運動のような低速かつ大きな斜航角を伴う操縦運動データは取得できない。そのため、本研究では低速操縦運動の効率的な計測を目的としてランダムにアクチュエータを操作する操船試験（以下、“ランダム操船試験”と称す）を実施し、その有効性を検証する。
- 過去の履歴を考慮するため先行研究では過去の全履歴を考慮する無限インパルス応答型 RNN [25, 96] または特定時点以前の記憶を無視する有限インパルス応答型 RNN [26] のいずれかが用いられてきたが、どちらがより操縦モデルの推定に適しているかを直接比較した例はない。そのため、本研究では、無限インパルス応答型 RNN と有限インパルス応答型 RNN の比較実験結果を示す。

この内容は、自著論文 1 [28] で発表されている内容に該当し、本学位論文では第 3 章で詳述される。

### (2) ANN を用いた操縦モデル推定のためのデータ拡張手法

操縦モデルの推定精度向上のため、本研究では加速度を用いないパラメータ訓練手法や操縦運動データ収集のためのランダム操船試験の導入などの提案している。しかし、ANN を用いたノンパラメトリック SI は、流体力学的背景を持たないため、操縦モデルの精度がデータの分布と質に依存しやすい。港内操船には標準操船だけでなく、低速操船時のさまざまなアクチュエータ操作や運動パターンが含まれ、結果として、取り得る状態空間の領域が広がり必要な運動データも増加する。一方、実船試験には高いコストがかかる。既存船の場合、日々のオペレーションデータが主要なデータ源となる可能性があるが、決まったパターンに従って操船される場合、データ分布に偏りが生じやすい。そのため、ランダム操船試験を用いたとしても、港内操船におけるあらゆる状況を網羅する操縦運動データを収集することは現実的に難しく、収集されるデータの内容はユーザにより異なり、必ずしも十分な操縦運動データが収集可能であるとは限らない。

そこで、本研究では操縦モデルの汎化性能向上のため、データ拡張 (Data Augmentation: DA) 手法の適用を検討する。DA 手法は、既存データから合成データを生成することで、訓練に使用するデータ量を増加させることが可能である。時系列データに適用する DA 手法については多くの研究 [97, 98] が存在するが、船舶の操縦モデル推定における適用例は知る限り存在しない。DA 手法には様々な種類があり、すべての手法が操縦モデルの汎化性能向上に寄与するわけではない。例えば、window wrapping [99–101] は、時系列データを圧縮または伸長して合成データを生成する手法であるが、この方法では操縦モデルによって予測される状態変数の時間微分が大きく変動する可能性があり、本研究には適していない。こうした DA 手法は元のデータとは異なる特性を持つデータを生成するため、かえって汎化性能を劣化させる恐れがある。

したがって、本研究の目的の一つは、ANN を用いた着桟操船および港内操船のための操縦モデル推定に DA 手法を導入し、その汎化性能を向上させることである。そのため、データ補強手法としてスライシング (slicing) 法とジッタリング (jittering) 法を採用し、効果的なデータ補強手法とその有効性を示す。この内容は自著論文 2 [33] で発表した内容に該当し、本学位論文では第 4 章で詳述される。

### (3) RL を用いた軌道追従制御方策の獲得手法

1.2.2 節で述べたように、RL は与えられた環境制約および不確実な状況下で最適な制御方策を見つけることが可能で、軌道追従制御問題へも適用可能である。しかし、最適なポリシーを獲得するためには RL において探索が必要であり、 $\epsilon$ -greedy 法 [94] や行動にランダムノイズ [102] を付加する方法が採用されるが、こうした探索方法では物理環境における安全性確保が難しい場合がある。また、トレーニングに必要な時間は環境や問題の複雑さに依存し、長時間を要することもあるため、物理環境における探索は安全性と経済性の観点から実用的でない可能性がある。

実環境での探索を避ける方法の一つとして、事前に強化学習のための操縦シミュレーション環境を構築するモデルベース強化学習 [103] が挙げられる。そのため、本研究では、MMG モデルに基づく操縦シミュレーション環境、または、操縦運動データから ANN で推定された操縦モデルに基づく操縦シミュレーション環境を構築し、着桟操船および港内操船の追跡制御方策に対応する RL 訓練手法を提案する。

RL による追従制御のための ANN 訓練に関する先行研究では、直線経路 [95] および曲線経路 [90] に対する訓練が実施されてきたが、港内操船や着桟操船に適応するためには、旋回、停止、後進、横移動など多様な操縦運動軌道を追従できるように訓練する必要がある。さらに、岸壁付近の障害物に接近する着桟操船では、わずかな追従誤差が衝突リスクを引き起こす可能性があるが、環境外乱を含む不確実な環境下では追従誤差を

完全にゼロに保つことは困難である。そのため、少なくとも衝突リスクのある追従誤差を回避することが有効であると考えられる。これらの課題解決のため、提案手法には以下の工夫が含まれている。

- 港内操船でとり得るさまざまな操縦運動軌道に対して訓練が行われるように、参照軌道生成のためのランダム操船を導入する。
- 障害物に対する衝突回避の訓練を可能にするために、参照軌道に応じた静的な擬似障害物の生成方法を提案する。
- 障害物と船舶間の距離を表す尺度を導入し、衝突を引き起こす可能性のある追従誤差回避させる報酬関数を提案する。

この内容は自著論文3 [104] および自著論文4 [105] で発表した内容に該当し、本学位論文では第5章で詳述される。

## 1.4 自著論文（学術雑誌掲載論文および学会論文）

本学位論文に含まれる内容に関する自著論文を以下にまとめる。

### 自著論文1 [28]

題名: On neural network identification for low-speed ship maneuvering model

著者名: Kouki Wakita, Atsuo Maki, Naoya Umeda, Yoshiki Miyauchi, Tohga Shimoji, Dimas M Rachman, Youhei Akimoto

出版物の種類: 学術雑誌 (全文査読付)

出版物の名称: Journal of Marine Science and Technology

出版社: Springer Nature

出版年月: 2022年3月1日

### 自著論文2 [33]

題名: Data Augmentation Methods of Dynamic Model Identification for Harbor Maneuvers using Feedforward Neural Network

著者名: Kouki Wakita, Yoshiki Miyauchi, Youhei Akimoto, Atsuo Maki

出版物の種類: 学術雑誌 (全文査読付)

出版物の名称: Journal of Marine Science and Technology

出版社: Springer Nature

出版年月: 承認済み (2024年10月27日)

### 自著論文3 [104]

題名: Collision probability reduction method for tracking control in automatic docking/berthing using reinforcement learning

著者名: Kouki Wakita, Youhei Akimoto, Dimas M Rachman, Yoshiki Miyauchi, Atsuo Maki

出版物の種類: 学術雑誌 (全文査読付)

出版物の名称: Journal of Marine Science and Technology

出版社: Springer Nature

出版年月: 2023年12月1日

**自著論文 4 [105]**

題名: Model-based reinforcement learning for trajectory tracking control of autonomous surface ship

著者名: Kouki Wakita

出版物の種類: 国際会議論文集 (全文査読付)

出版物の名称: Proceedings of the 34th International Society of Offshore and Polar Engineers

出版年月: 2024 年 6 月 16 日

## 第 2 章

### 事前知識

本章では、本論文の前提となる数学的表記法および船舶の操縦運動に関する基礎知識について述べる。また、提案手法の検証実験に使用した供試模型船についても詳述する。以降、2.1 節で数学的表記法について説明し、2.2 節では船舶の操縦運動に関する座標系および運動方程式について述べる。さらに、2.3 節及び2.4 節で本研究に使用された二隻の供試模型船について詳述し、最後に2.5 節で供試船を用いた試験が実施された実験池について述べる。

#### 2.1 表記法

本節では、本論文を通して用いる数学的表記法についてまとめる。 $\mathbb{R}$  は実数の集合、 $\mathbb{S} = [0, 2\pi]$  は角度の集合を表す。 $n$  次元のユークリッド空間は  $\mathbb{R}^n$  と表記される。ある実数ベクトルを  $\mathbf{x} \in \mathbb{R}^n$ 、実数行列を  $\mathbf{A} \in \mathbb{R}^{n \times n}$  とすると、 $\|\mathbf{x}\|$  は  $\mathbf{x}$  のユークリッドノルム  $(\mathbf{x}^T \mathbf{x})^{1/2}$ 、 $\|\mathbf{x}\|_{\mathbf{A}}$  は  $\mathbf{A}$  で重み付けされた  $\mathbf{x}$  のユークリッドノルム  $(\mathbf{x}^T \mathbf{A} \mathbf{x})^{1/2}$  を表す。また、 $\text{diag}(\mathbf{x}) \in \mathbb{R}^{n \times n}$  は  $\mathbf{x}$  を対角要素を持つ対角行列、 $\text{arctan2}$  は二つの引数を取る逆正接関数であり、 $\text{arctan2}(x, y)$  は  $xy$  直交座標における点  $(x, y)$  の偏角を示す。さらに、 $\mathbf{1}_a$  は  $a$  が真であれば 1、そうでなければ 0 を取る指示関数を表し、 $\lfloor \cdot \rfloor$  は床関数を表す。

#### 2.2 平水中における 3 自由度の操縦運動

本研究では、主に港湾内の操縦運動および操船制御に焦点を当てているため、風外乱を考慮した穏やかな水面における 3 自由度の操縦運動を扱う。港湾内であっても、船舶の操縦運動は波や潮流などの環境外乱に影響を受ける可能性があるが、これらの外乱状態は無視されるか、観測されない変数として扱われる。本節では、2.2.1 節で船舶の操縦運動を表現するための座標系や変数の定義について詳述し、2.2.2 節で船舶の操縦運動を表現する操縦運動方程式について述べる。

##### 2.2.1 座標系

Fig. 2.1に示されるように、空間固定座標系  $O_0-x_0y_0z_0$  と、ミッドシップを原点とする船体固定座標系  $O-xyz$  を考える。船舶の運動は、姿勢ベクトル  $\boldsymbol{\eta} \equiv (x_0, y_0, \psi)^T \in \mathbb{R}^2 \times \mathbb{S}$  および速度ベクトル  $\boldsymbol{v} \equiv (u, v_m, r)^T \in \mathbb{R}^3$  によって表現される。ここで、 $(x_0, y_0)$  は地球固定座標系  $O_0 - x_0y_0z_0$  におけるミッドシップの位置、 $\psi$  は回頭角、 $(u, v_m)$  はそれぞれ船体の縦方向および横方向の速度、 $r$  は回頭角速度を意味する。さらに、簡略化のため、船の運動状態変数をまとめて  $\mathbf{x} \equiv (\boldsymbol{\eta}^T, \boldsymbol{v}^T)^T$  と定義する。

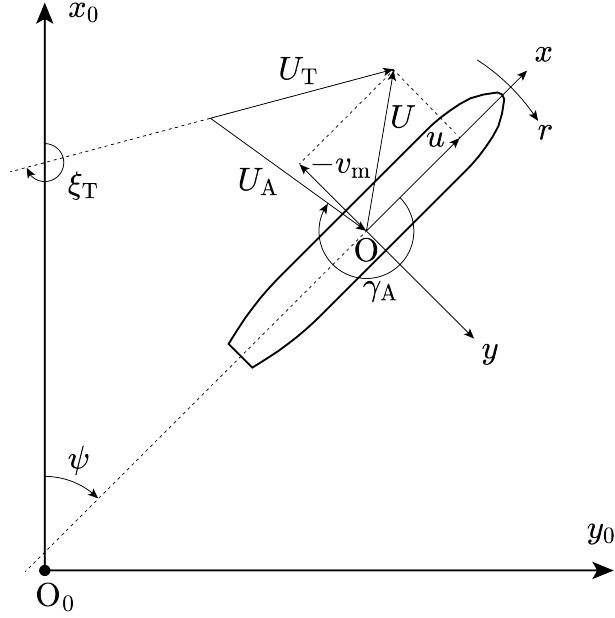


Fig. 2.1: Coordinate Systems of ship maneuvering motion.

船舶にはプロペラや舵といったアクチュエータによって外力が与えられる。船舶に搭載されるアクチュエータは船種によって異なるが、本研究では主にプロペラ、舵、サイドスラスターを考慮する。すなわち、アクチュエータ状態はプロペラ回転数、舵角、サイドスラスター回転数で表される。アクチュエータ状態の詳細な定義については、2.3 節及び2.4 節においてそれぞれの供試船に応じて述べるが、本節では  $N_u$  次元のアクチュエータ状態ベクトルを  $\mathbf{u} \in \mathcal{U}$  と表記する。ここで、アクチュエータ状態  $\mathbf{u}$  の  $i$  番目の成分の上下限をそれぞれ  $u_i$ ,  $\bar{u}_i$  として、 $\mathcal{U} = \prod_i^{N_u} [u_i, \bar{u}_i]$  と定義する。

次に、船舶が風から受ける力を考慮する。風の状態は、真風速および真風向を成分とするベクトル  $\mathbf{w}_T \equiv (U_T, \xi_T)^T \in \mathbb{R} \times \mathbb{S}$  で表され、 $U_T$  は真風速、 $\xi_T$  は真風向を示す。ここで、真風向の基準方向は風が  $x_0$  軸の正の方向から負の方向に吹く向きと定義される。本研究では、真風速および真風向は時刻  $t$  に応じて変動するものの、空間的には均一であると仮定する。また、船舶が直接受ける影響は真風ではなく相対風によるものであり、相対風の状態は相対風速および相対風向を成分とするベクトル  $\mathbf{w}_A \equiv (U_A, \gamma_A)^T \in \mathbb{R} \times \mathbb{S}$  で表される。ここで、 $U_A$  は相対風速、 $\gamma_A$  は相対風向を示し、相対風向の基準方向は風が船首から船尾に向かって吹く向きと定義される。

真風と相対風は、ミッドシップの速度  $(u, v_m)$  と回頭角  $\psi$  が与えられれば、相互に変換可能である。船舶固定座標系における真風状態ベクトルを  $\mathbf{w}'_T \equiv (U_T \cos(\xi_T - \psi), U_T \sin(\xi_T - \psi))^T \in \mathbb{R}^2$ 、相対風状態ベクトルを  $\mathbf{w}'_A \equiv (U_A \cos \gamma_A, U_A \sin \gamma_A)^T \in \mathbb{R}^2$  とそれぞれ定義し、船舶の速度ベクトル  $\mathbf{v} \equiv (u, v_m)^T \in \mathbb{R}^2$  と定義すると、真風と相対風の関係は

$$\mathbf{w}'_A = \mathbf{w}'_T - \mathbf{v} \quad (2.2.1)$$

と表される。よって、船の速度  $(u, v_m)$  と回頭角  $\psi$  が与えられれば、相対風速  $U_A$  および相対風向  $\gamma_A$  は

$$\begin{cases} U_A = \sqrt{(U_T \cos(\xi_T - \psi) - u)^2 + (U_T \sin(\xi_T - \psi) - v_m)^2} \\ \gamma_A = \arctan2(U_T \sin(\xi_T - \psi) - v_m, U_T \cos(\xi_T - \psi) - u) \end{cases} \quad (2.2.2)$$

と表現される。

### 2.2.2 操縦運動方程式

本研究では、船舶の操縦運動は主に船体の運動学的変数ベクトル  $\mathbf{x}$ 、アクチュエータ状態変数ベクトル  $\mathbf{u}$ 、真風速風向ベクトル  $\mathbf{w}_T$  によって決定されると仮定する。しかし、実際には波や潮流といった環境外乱や、斜航角が大きい低速操縦運動における渦に起因する流体力の変動などの影響を受ける可能性があり、船体の運動学的変数ベクトル  $\mathbf{x}$ 、アクチュエータ状態変数ベクトル  $\mathbf{u}$ 、真風速風向ベクトル  $\mathbf{w}_T$  のみでは船舶の操縦運動を必ずしも十分に説明できないことがある。一般に、渦や潮流の計測には特別な計測機器が必要であるため、本研究ではこれらは観測されないものとした。このように観測可能な変数が限られる部分観測問題では、観測不可能な現在の状態を過去の観測履歴から推定する必要が生じる可能性がある。

従って、現在時刻を含む過去  $N_m$  ステップ間の離散時刻における状態変数  $\mathbf{x}$ ,  $\boldsymbol{\eta}$ ,  $\mathbf{v}$ ,  $\mathbf{u}$ ,  $\mathbf{w}_T$ ,  $\mathbf{w}_A$  をまとめたベクトルをそれぞれ

$$\left\{ \begin{array}{l} \mathbf{X}_{N_m}^{(-)}(t) \equiv \left( \mathbf{x}^T(t), \mathbf{x}^T(t - \Delta t), \dots, \mathbf{x}^T(t - (N_m - 1)\Delta t) \right)^T \in \mathbb{R}^{6N_m} \\ \mathbf{H}_{N_m}^{(-)}(t) \equiv \left( \boldsymbol{\eta}^T(t), \boldsymbol{\eta}^T(t - \Delta t), \dots, \boldsymbol{\eta}^T(t - (N_m - 1)\Delta t) \right)^T \in \mathbb{R}^{3N_m} \\ \mathbf{N}_{N_m}^{(-)}(t) \equiv \left( \mathbf{v}^T(t), \mathbf{v}^T(t - \Delta t), \dots, \mathbf{v}^T(t - (N_m - 1)\Delta t) \right)^T \in \mathbb{R}^{3N_m} \\ \mathbf{U}_{N_m}^{(-)}(t) \equiv \left( \mathbf{u}^T(t), \mathbf{u}^T(t - \Delta t), \dots, \mathbf{u}^T(t - (N_m - 1)\Delta t) \right)^T \in \mathbb{R}^{N_u N_m} \\ \mathbf{W}_{T,N_m}^{(-)}(t) \equiv \left( \mathbf{w}_T^T(t), \mathbf{w}_T^T(t - \Delta t), \dots, \mathbf{w}_T^T(t - (N_m - 1)\Delta t) \right)^T \in \mathbb{R}^{2N_m} \\ \mathbf{W}_{A,N_m}^{(-)}(t) \equiv \left( \mathbf{w}_A^T(t), \mathbf{w}_A^T(t - \Delta t), \dots, \mathbf{w}_A^T(t - (N_m - 1)\Delta t) \right)^T \in \mathbb{R}^{2N_m}, \end{array} \right. \quad (2.2.3)$$

と定義する。ただし、 $\Delta t$  はタイムステップを表す。このとき、船舶の操縦運動は常微分方程式

$$\dot{\boldsymbol{\eta}} = \mathbf{R}(\boldsymbol{\eta}) \mathbf{v}, \quad (2.2.4a)$$

$$\dot{\mathbf{v}} = \mathbf{F} \left( \mathbf{N}_{N_m}^{(-)}, \mathbf{U}_{N_m}^{(-)}, \mathbf{W}_{A,N_m}^{(-)} \right) \quad (2.2.4b)$$

で表現可能であると仮定される。ただし、 $\mathbf{F}$  は操縦モデルを表す未知のベクトル関数、 $\mathbf{R}(\boldsymbol{\eta}) \in \mathbb{R}^{3 \times 3}$  は  $O - xyz$  から  $O_0 - x_0y_0z_0$  への回転行列であり

$$\mathbf{R}(\boldsymbol{\eta}) = \begin{bmatrix} \cos \psi & -\sin \psi & 0 \\ \sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.2.5)$$

と定義される。表記の簡略化のため、以降では、Eq. (2.2.4b) と Eq. (2.2.2) の合成関数を

$$\dot{\mathbf{v}} = \mathbf{F}' \left( \mathbf{X}_{N_m}^{(-)}, \mathbf{U}_{N_m}^{(-)}, \mathbf{W}_{T,N_m}^{(-)} \right) \quad (2.2.6)$$

と表記し、また、Eq. (2.2.4a) や Eq. (2.2.6) はまとめて

$$\dot{\mathbf{x}} = \mathbf{f} \left( \mathbf{X}_{N_m}^{(-)}, \mathbf{U}_{N_m}^{(-)}, \mathbf{W}_{T,N_m}^{(-)} \right) \quad (2.2.7)$$

と表記する。ただし、Eq. (2.2.4b) がマルコフ特性を持つと仮定する場合は、 $N_m = 0$  となり  $\mathbf{X}_{N_m}^{(-)}$ ,  $\mathbf{H}_{N_m}^{(-)}$ ,  $\mathbf{N}_{N_m}^{(-)}$ ,  $\mathbf{U}_{N_m}^{(-)}$ ,  $\mathbf{W}_{T,N_m}^{(-)}$ ,  $\mathbf{W}_{A,N_m}^{(-)}$  はすべて  $\mathbf{x}$ ,  $\boldsymbol{\eta}$ ,  $\mathbf{v}$ ,  $\mathbf{u}$ ,  $\mathbf{w}_T$ ,  $\mathbf{w}_A$  に置き換わることに注意する。



Fig. 2.2: Subject ship 1 (M.V. Esso Osaka).

Table 2.1: Actuator state limitations of subject ship 1.

Item	Symbol	Range
Rudder angle (degree)	$\delta$	$[-35^\circ, 35^\circ]$
Propeller revolution (rps)	$n_P$	$[-20, 20]$

## 2.3 供試船 1：一軸一舵を搭載した模型船

本研究では異なる二つの供試模型船が利用された。本節では、一つ目の供試船について詳述する。一つ目の供試船は、Fig. 2.2に示す、M.V. Esso Osaka の 3 メートル模型船である。この模型船は、一つのプロペラと一枚の舵が搭載されている。よって、この船のアクチュエータ状態変数は舵角およびプロペラ回転数であり、それぞれ、 $\delta$  および  $n_P$  と表記する。アクチュエータ状態変数のとり得る値を Table 2.1に示す。

### 2.3.1 自由航走試験のため計測および操船システム

この模型船には、和田らによって開発された自由航走試験のための操船システムが搭載されており、Robot Operating System (ROS)に基づいて動作する [106]。本研究では、この模型船を用いて操縦運動データ集合の計測および操船制御実験を行った。

この模型船には、光ファイバージャイロ (Fiber Optic Gyroscope: FOG) 1 台、全球測位衛星システム (Global Navigation Satellite System: GNSS) 受信機 3 台、および超音波風速計 2 台が搭載されており、これらの計測値を基に船体状態  $\mathbf{x}$  および風状態  $\mathbf{w}_A$  が観測される。

船体位置  $(x_0, y_0)$  は、GNSS で得られる位置情報をミッドシップ位置へ変換することで観測しており、GNSS にはセンチメートル級測位補強サービス (Centimeter Level Augmentation Service: CLAS) を用いた精密な位置測定が採用されている。船体速度  $(u, v_m)$  は、ミッドシップ位置の数値時間微分から計測され、その後、線形カルマンフィルタで平滑化を行った。回頭角速度  $r$  は光ファイバージャイロで直接計測し、回頭角  $\psi$  はドリフトの影響を避けるため、FOG ではなく、複数の GNSS 受信機の相対位置から算出している。さらに、舵角  $\delta$  およびプロペラ回転数  $n_P$  も逐次計測しており、これらすべての値は 10 Hz の周波数でサンプリングされている。

### 2.3.2 操縦シミュレーションのための MMG モデル

操縦運動データの生成および操縦シミュレーション環境として使用する供試船1のMMGモデルについて述べる。本研究は港内操船に焦点を当てているため、斜航角を大きくとる運動やプロペラ逆転を考慮したサブモデルが選択されている。MMGモデルの係数は経験式もしくは拘束模型試験により推定された。

標準的なMMGモデルで使用される3自由度の運動方程式は

$$\begin{cases} (m + m_x) \dot{u} - (m + m_y) v_m r - x_G m r^2 = X \\ (m + m_y) \dot{v}_m + (m + m_x) u r + x_G m \dot{r} = Y \\ (I_{zz} + J_{zz} + x_G^2 m) \dot{r} + x_G m (\dot{v}_m + u r) = N \end{cases} \quad (2.3.1)$$

と表される。ここで、 $m$ ,  $m_x$ ,  $m_y$  はそれぞれ船体質量、Surge 方向の付加質量、Sway 方向の付加質量を表し、 $I_{zz}$ ,  $J_{zz}$  はそれぞれ慣性モーメント、付加慣性モーメントを表し、 $x_G$  は Surge 方向の重心位置を表す。Eq. (2.3.1) の右辺の  $X$ ,  $Y$ ,  $N$  はそれぞれ縦力、横力、Yaw 方向のモーメントを表す。これらの力およびモーメントは船体、プロペラ、舵、風に起因する力およびモーメントに分解され、

$$\begin{cases} X = X_H + X_P + X_R + X_A \\ Y = Y_H + Y_P + Y_R + Y_A \\ N = N_H + N_P + N_R + N_A \end{cases} \quad (2.3.2)$$

と表される。ここで、下付き文字の H, P, R, A はそれぞれ船体、プロペラ、舵、風に起因する力であることを示している。これらの力についてそれぞれ以降で詳述する。

#### 2.3.2.1 船体に起因する力およびモーメント

まず、船体に働く力およびモーメントは、芳村ら [8] が提案した外洋および港湾操船の統一数理モデルを用いて表現する。つまり、船体に働く力およびモーメントは

$$\begin{cases} X_H = \left(\frac{\rho}{2}\right) L_{pp} d \left[ \left\{ X'_{0(F)} + \left( X'_{0(A)} - X'_{0(F)} \right) (\beta/\pi) \right\} u U + X'_{vr} L_{pp} \cdot v_m r \right] \\ Y_H = \left(\frac{\rho}{2}\right) L_{pp} d \left[ Y'_v v_m |u| + Y'_r L_{pp} \cdot r u - \left( \frac{C_D}{L_{pp}} \right) \int_{-L_{pp}/2}^{L_{pp}/2} |v_m + rx| (v_m + C_{rY} rx) dx \right] \\ N_H = \left(\frac{\rho}{2}\right) L_{pp}^2 d \left[ N'_v v_m u + N'_r L_{pp} \cdot r |u| - \left( \frac{C_D}{L_{pp}^2} \right) \int_{-L_{pp}/2}^{L_{pp}/2} |v_m + C_{rN} rx| (v_m + C_{rN} rx) x dx \right] \end{cases} \quad (2.3.3)$$

と表現される。ただし、 $\rho$  は水の密度、 $L_{pp}$  は船の垂線間長、 $d$  は喫水、 $X'_{0(F)}$  および  $X'_{0(A)}$  はそれぞれ前進および後進の抵抗係数、 $C_D$  は横流れ抵抗力係数、 $C_{rY}$  および  $C_{rN}$  は横力と Yaw モーメントの補正係数、 $X'_{0(F)}$ ,  $Y'_v$ ,  $Y'_r$ ,  $N'_v$ , および  $N'_r$  はそれぞれ無次元の流体力微係数である。Eq. (2.3.3) に本来含まれるはずであった付加質量項は Eq. (2.3.1) に移動されていることに注意する。抵抗係数と線形流体力微係数はキャプティブモデル試験により決定され、残りの係数は経験式 [8] を用いて推定された。

#### 2.3.2.2 プロペラに起因する力およびモーメント

本研究では、後進運動やプロペラ逆転が取られることが想定される。そのため、標準 MMG モデル [3] で使用されるプロペラモデルに加えて、前進速度およびプロペラ回転数に応じたサブモデルが追加されている。

(1)  $n_P > 0$  のとき

プロペラに起因する Surge 方向の力は

$$X_p = \rho n_p^2 D_p^4 (1 - t_p) K_T \quad (2.3.4)$$

と表現される。ここで、 $K_T$  は前進係数  $J_p = \frac{(1-w_p)u}{n_p D_p}$  の多項式で表現される推力係数である。また、 $w_p$  は有効プロペラ伴流係数であり、

$$1 - w_p = 1 - w_{p0} + \tau |v'_m + x'_p r'| + C'_p (v'_m + x'_p r')^2 \quad (2.3.5)$$

と計算される [107]。ここで、 $w_{p0}$  は  $v_m = r = 0$  における伴流率、 $\tau$ 、 $C'_p$ 、 $x'_p$  は経験係数である。推力減少係数  $t_p$  および伴流率  $w_p$  はプロペラ条件によって異なるが、先行研究 [6, 108] と同様に

$$\begin{cases} t_p = 0 & \text{for } n_p < 0 \\ w_p = 0 & \text{for } u < 0 \end{cases} \quad (2.3.6)$$

と定義された。また、プロペラに起因する横力と Yaw モーメントは標準 MMG モデルでは無視されるが、上野ら [109] の拘束模型試験に基づく多項式を参考に

$$\begin{cases} Y_p = \begin{cases} 0 & \text{for } u \geq 0 \\ \frac{1}{2} \rho L_{pp}^2 d (n_p P)^2 (A_6 J_s^2 + A_7 J_s + A_8) & \text{for } u < 0 \end{cases} \\ N_p = \begin{cases} 0 & \text{for } u \geq 0 \\ \frac{1}{2} \rho L_{pp}^2 d (n_p P)^2 (B_6 J_s^2 + B_7 J_s + B_8) & \text{for } u < 0 \end{cases} \end{cases}, \quad (2.3.7)$$

と表現する。ここで、 $P$  はプロペラピッチ、 $J_s = \frac{u}{n_p D_p}$ 、 $A_6, A_7, A_8$  and  $B_6, B_7, B_8$  は多項式係数である。

(2)  $n_P < 0$  のとき

プロペラに起因する Surge および Sway 方向の力、Yaw モーメントは、拘束模型試験 [110] に基づいて

$$\begin{cases} X_p = \rho n_p^2 D_p^4 \begin{cases} C_6 + C_7 J_s & \text{for } (J_s \geq C_{10}) \\ C_3 & \text{for } (J_s < C_{10}) \end{cases} \\ Y_p = \frac{1}{2} \rho L d (n_p D_p)^2 \begin{cases} A_1 + A_2 J_s & (-0.35 \leq J_s \leq -0.06) \\ A_3 + A_4 J_s & (J_s < -0.35) \\ A_5 & (-0.06 < J_s) \end{cases} \\ N_p = \frac{1}{2} \rho L^2 d (n_p D_p)^2 \begin{cases} B_1 + B_2 J_s & (-0.35 \leq J_s \leq -0.06) \\ B_3 + B_4 J_s & (J_s < -0.35) \\ B_5 & (-0.06 < J_s) \end{cases} \end{cases} \quad (2.3.8)$$

と表現した。ここで、 $A_1, A_2, \dots, A_5, B_1, B_2, \dots, B_5, C_3, C_6, C_7$  は多項式係数である。

## 2.3.2.3 舵に起因する力およびモーメント

本研究では、後進運動やプロペラ逆転が取られることが想定される。そのため、標準 MMG モデル [3] で使用されるプロペラモデルに加えて、前進速度およびプロペラ回転数に応じたサブモデルが追加されている。標

準 MMG モデル [3] では、舵のモデルは

$$\begin{cases} X_R = -(1 - t_R) F_N \sin \delta \\ Y_R = -(1 - a_H) F_N \cos \delta \\ N_R = -(x_R + a_H x_H) F_N \cos \delta \end{cases} \quad (2.3.9)$$

と表される。ここで、 $t_R$  は舵による推力減少係数、 $x_R$  はミッドシップからの舵の Surge 方向の位置、 $a_H$  は横力の補正係数、 $x_H$  は舵による船体への横力の作用点の位置、 $F_N$  は舵直圧力は

$$F_N = (1/2)\rho A_R U_R^2 f_\alpha \sin \alpha_R \quad (2.3.10)$$

と定義される。ただし、 $A_R$  は舵の面積、 $U_R$  は舵流入速度、 $f_\alpha$  は舵直圧力勾配、 $\alpha_R$  は有効流入角である。接岸操作では低船速でも舵が動作するため、舵直圧力勾配  $f_\alpha$  は一定ではなくなる。しかし、本研究では、 $f_\alpha$  は一定と仮定し、藤井の式 [111]、

$$f_\alpha = 6.13\Lambda / (2.25 + \Lambda) \quad (2.3.11)$$

によって決定される。ここで  $\Lambda$  は舵アスペクト比を表す。

次に、舵流入速度  $U_R$  と有効流入角  $\alpha_R$  は、縦方向の流入速度  $u_R$  と横方向の流入速度  $v_R$  をもちいて、

$$\begin{cases} U_R = \sqrt{u_R^2 + v_R^2} \\ \alpha_R = \delta - \arctan(u_R, v_R) \end{cases} \quad (2.3.12)$$

と表される。ここで、横方向の流入速度  $v_R$  は、整流係数  $\gamma_P, \gamma_N$  と実験定数  $l_R$  を用いて、

$$v_R = \begin{cases} -\gamma_P(v_m + l_R r) & \text{for } v_m + x_R r \geq 0 \\ -\gamma_N(v_m + l_R r) & \text{for } v_m + x_R r < 0 \end{cases} \quad (2.3.13)$$

と表される。また、縦方向の流入速度  $u_R$  は船の進行方向とプロペラに誘起された流れの影響を大きく受けるため、 $n_P \geq 0, u_R$  の時、低速域が修正された式 [112]、すなわち、

$$u_R = \varepsilon \sqrt{\eta \left\{ u_p + \frac{k_x}{\varepsilon} \left( \sqrt{u_p^2 + \frac{8KT(n_p D_p)^2}{\pi}} - u_p \right) \right\}^2 + (1 - \eta) u_p^2} \quad (2.3.14)$$

と表現される。ただし、 $u_p = (1 - w_p)u$ 、 $\eta = D_p/H_R$ 、 $H_R$  は舵の高さ、 $\varepsilon$  は後流割合の比、 $k_x$  は経験係数である。 $u \geq 0, n_p < 0$  のときは、北川のモデル [113]、すなわち、

$$u_R = \operatorname{sgn}(u_{Rsq}) \cdot \sqrt{|u_{Rsq}|} \quad (2.3.15)$$

と表現される。ただし、

$$\begin{cases} u_{Rsq} = \eta \cdot \operatorname{sgn}(u_{RPR1}) \cdot u_{RPR1}^2 + (1 - \eta) \operatorname{sgn}(u_{RPR2}) \cdot u_{RPR2}^2 + C_{PR} \cdot u \\ u_{RPR1} = u \varepsilon (1 - w_p) + n_p D_p k_{xPR} \sqrt{8|KT|/\pi} \\ u_{RPR2} = u \varepsilon (1 - w_p) \end{cases} \quad (2.3.16)$$

ここで、 $k_{xPR}$  と  $C_{PR}$  はそれぞれ速度増加係数とプロペラ反転条件の補正係数である。また、 $u < 0, n_p < 0$  では、流入は船の運動と等しいと仮定した、つまり、 $u_R = u$  とした [6]。

Table 2.2: Limitations of the actuator state variables.

Item	Symbol	Range
Port side rudder angle (degree)	$\delta_P$	$[-105^\circ, 35^\circ]$
Starboard side rudder angle (degree)	$\delta_S$	$[-35^\circ, 105^\circ]$
Propeller revolution (rps)	$n_P$	$[0, 20]$
Bow thruster revolution (rps)	$n_{BT}$	$[-30, 30]$

### 2.3.2.4 風に起因する力およびモーメント

風に起因する力およびモーメントは、藤原の回帰式 [15]、つまり、

$$\begin{cases} X_A = (1/2)\rho_A U_A^2 A_T \cdot C_X \\ Y_A = (1/2)\rho_A U_A^2 A_L \cdot C_Y \\ N_A = (1/2)\rho_A U_A^2 A_{LLOA} \cdot C_N \end{cases} \quad (2.3.17)$$

と表現される。ただし、

$$\begin{aligned} C_X &= X_0 + X_1 \cos(2\pi - \gamma_A) + X_3 \cos 3(2\pi - \gamma_A) + X_5 \cos 5(2\pi - \gamma_A) \\ C_Y &= Y_1 \sin(2\pi - \gamma_A) + Y_3 \sin 3(2\pi - \gamma_A) + Y_5 \sin 5(2\pi - \gamma_A) \\ C_N &= N_1 \sin(2\pi - \gamma_A) + N_2 \sin 2(2\pi - \gamma_A) + N_3 \sin 3(2\pi - \gamma_A) . \end{aligned} \quad (2.3.18)$$

ここで、 $\rho_A$  は空気の密度、 $A_T$ 、 $A_L$ 、 $L_{OA}$  はそれぞれ船の横断面積、船の横投影面積、船の全長である。 $X_i, Y_i, N_i$  は船舶の幾何学的パラメータを説明変数とする風圧係数であり、これらの係数は多数の縮尺船舶模型の風洞試験データに基づいて導出された [15]。

## 2.4 供試船 2: VecTwin 舵システムを搭載した模型船

二つ目の供試船は、Fig. 2.3に示す、VecTwin 舵システムとバウスラスターを搭載した一軸の 3 メートル模型船である。VecTwin 舵システム [114] は、一つの固定ピッチプロペラ (Fixed Pitch Propeller: FPP) の後ろに二つのフィッシュテール舵が搭載され、それぞれの舵が個別に動作することで高い操縦性能を実現する [115]。このシステムは、少なくとも 1 つのサイドスラスターがあれば、プロペラを一定の回転数で正転させたまま、ホバリングやクラビング、後進、回転といった特殊な操船が可能である [87, 116]。

この船のアクチュエータ状態変数は、左舷 (Port) 側の舵角、右舷 (Starboard) 側の舵角、プロペラ回転数、バウスラスターの回転数であり、それぞれ  $\delta_P, \delta_S, n_P, n_{BT}$  と表記する。これらのアクチュエータ状態変数のとり得る値は Table 2.2に示す。

### 2.4.1 自由航走試験のため計測および操船システム

この模型船にも、2.3 節で述べた供試船 1 と同様に、自由航走試験のための ROS に基づく操船システムが搭載されている。この模型船には、FOG が 1 台、GNSS 受信機が 3 台、超音波風速計が 1 台搭載され、これらの計測値を基に船体状態  $\mathbf{x}$  および風状態  $\mathbf{w}_A$  が観測される。

ただし、観測方法には2.3.1 節との間でいくつかの相違がある。FOG の方位計測精度が向上したため、回頭角  $\psi$  は光ファイバージャイロによって直接計測される。また、船体速度 ( $u, v_m$ ) は、GNSS 受信機で計測され



Fig. 2.3: Subject ship 2 (VecTwin rudder ship).

た対地速度 (Speed Over Ground: SOG), 対地針路 (Course Over Ground: COG), および FOG で計測された回頭角  $\psi$  を基に算出される。これらのすべてのデータは 10 Hz の周波数で計測されている。

#### 2.4.2 操縦シミュレーションのための MMG モデル

供試船 2 の MMG モデルについて述べる。本研究では、供試船 1 と同様に、港内操船に焦点が当てられているため、斜航角を大きく取る運動やプロペラ逆転などに対応したサブモデルが選択されている。MMG モデルの係数はそれぞれ、経験式もしくは拘束模型試験によって推定された。

供試船 2 では、Eq. (2.3.1) の右辺の  $X$ ,  $Y$ ,  $N$  は船体, プロペラ, 舵, 風に起因する力およびモーメントに分解され,

$$\begin{cases} X = X_H + X_P + X_R + X_{BT} + X_A \\ Y = Y_H + Y_P + Y_R + Y_{BT} + Y_A \\ N = N_H + N_P + N_R + n_{BT} + N_A \end{cases} \quad (2.4.1)$$

と表される。ここで、下付き文字の H, P, R, BT, A はそれぞれ船体, プロペラ, 舵, バウスラスタ, 風に起因する力であることを示している。船体流体力は、2.3.2.1 節で述べた芳村のモデル [8] を用い、抵抗係数と線形流体力学的導関数はキャプティブモデル試験により決定され、残りの係数は経験式 [8] を用いて推定された。舵とプロペラに起因する力は Kang のモデル [117] が使用され、プロペラの推力係数はプロペラ単独試験から、残りの係数は Kang [117] の VLCC の係数を適用した。2.3.2.4 節で述べた藤原の回帰式を用いて表された。そして、スラスタに起因する力およびモーメントについては以降で詳述する。

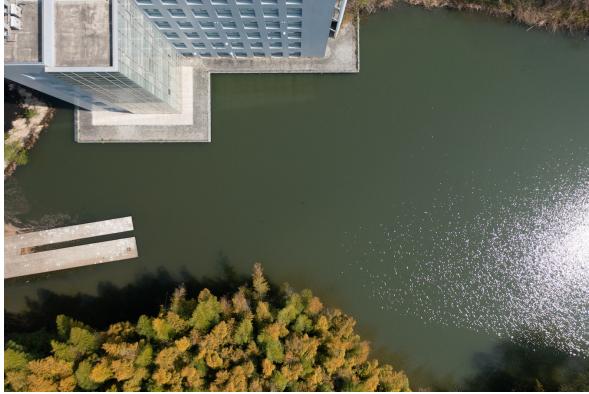


Fig. 2.4: Bird's-eye view of inukai pond.

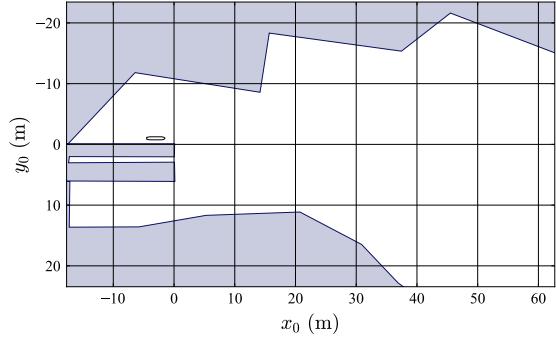


Fig. 2.5: Geometry of inukai pond and subject ship.

#### 2.4.2.1 スラスターに起因する力およびモーメント

スラスター力は Kobayashi のモデル [118] に基づき以下のようにモデル化した.

$$\begin{cases} X_{BT} = 0 \\ Y_{BT} = \left(1 + a_{YSB1} + a_{YSB2} \cdot Fr + a_{YSB3} \cdot Fr^2\right) \cdot T_{BT} \\ N_{BT} = \left(1 + a_{NSB1} + a_{NSB2} \cdot Fr + a_{NSB3} \cdot Fr^2\right) \cdot T_{BT} \cdot x_{BT} \\ T_{BT} = \rho D_{BT}^4 n_{BT}^2 K_{T,BT} \end{cases} \quad (2.4.2)$$

ただし,  $a_{YSB1}, a_{YSB2}, a_{YSB3}, a_{NSB1}, a_{NSB2}, a_{NSB3}$  はスラスター力の低減と船速の関係を表すフルード数に対する 2 次関数の係数で,  $x_{BT}$  はサイドスラスターの縦方向の位置である. これらの係数はキャプティブ模型試験により決定され, 推力係数  $K_{T,BT}$  は実験的に求めた.

## 2.5 実験池: 犬飼池

自由航走試験のために用いられた実験池について説明する. 本研究では, 操縦運動データ集合の収集および制御器の検証のために自由航走試験を実施するが, すべての自由航走試験は大阪大学実験池の犬飼池で実施されている. 犬飼池の鳥瞰図を Fig. 2.4 に示し, また, 犬飼池の形状および座標系を Fig. 2.5 に示す.

## 第3章

# 人工ニューラルネットワークを用いた操縦モデルの推定手法

本章では、ANNを用いた港内操船のための操縦モデルの推定手法について述べ、模型船の自由航走試験データを用いた検証実験結果を示す。3.1節では、順伝播型ニューラルネットワークや回帰型ニューラルネットワークの概要を説明し、それらを用いた操縦モデルの構造について述べる。3.2節では、操縦モデルの入出力関係を学習する回帰分析ベースの訓練手法と新たに導入する軌道推定ベースの手法についてそれぞれ述べる。3.3節で模型船の自由航走試験データを用いた操縦モデルの推定実験の結果を示し、3.4節で結果を元に考察を述べ、最後に3.5節では本章の結言について述べる。

### 3.1 操縦モデルの構造

本節では、ANNを用いた操縦モデルの構造について詳述する。ANNは人間の神経ネットワークを模擬した計算モデルである。ANNは、人工ニューロンと呼ばれる重み付け総和と活性化関数が組み合わされた計算ユニットで構成されており、ANNは多数の人工ニューロンが相互に接続されて構成されたネットワークである。本研究では順伝播型ニューラルネットワークと回帰型ニューラルネットワークが用いられる。それぞれのANNモデルについて3.1.1節及び3.1.2節で述べる。

#### 3.1.1 順伝播型ニューラルネットワーク

順伝播型ニューラルネットワーク(Feedforward Neural Network: FNN)は内部に循環を持たないANNで、機械学習タスクで頻繁に使用される代表的なANNモデルの一つである。FNNでは、多くの場合、複数の人工ニューロンのまとまりを一つの層と考えられる。その層間の結合方法に応じた様々なモデルが存在し、特に、全結合層のみで構成されるFNNは多層パーセプトロンと呼ばれる。FNNは、普遍的な近似能力を持つことがしられており[40, 41]、ミニバッチ学習により大規模なデータに対して学習が可能な[119, 120]、非常に便利で強力なツールである。

### 3.1.1.1 全結合層

全結合層では、層中のそれぞれの人工ニューロンが次の層の全ての人工ニューロンと結合している。つまり、 $l$  層目の潜在変数を  $\mathbf{z}_l \in \mathbb{R}^{N_{\mathbf{z}_l}}$  と表記すると、 $l$  層目から  $l+1$  層目への全結合層による変換は

$$\mathbf{z}_{l+1} = \mathbf{g}_l(\mathbf{W}_l \mathbf{z}_l + \mathbf{b}_l) \quad (3.1.1)$$

と定義される。ここで、 $\mathbf{W}_l \in \mathbb{R}^{N_{\mathbf{z}_{l+1}} \times N_{\mathbf{z}_l}}$  と  $\mathbf{b}_l \in \mathbb{R}^{N_{\mathbf{z}_{l+1}}}$  は全結合層に含まれる重み行列とバイアスベクトルを表し、 $\mathbf{g}_l$  は

$$\mathbf{g}_l(\mathbf{z}) = \left( g_l(z_1), g_l(z_2), \dots, g_l(z_{N_{\mathbf{z}_{l+1}}}) \right)^T \quad (3.1.2)$$

と定義され、 $g_l$  は活性化関数であり、双曲線正接関数 ( $\tanh$ ) や正規化線形関数 (ReLU) などがよく用いられる。つまり、全結合層では、 $l$  層目の潜在変数の重み付き和が取られたのちに、全ての変数が活性化関数により変換される。本論文では、この全結合層の処理を

$$\mathbf{z}_{l+1} = \mathbf{h}_l^{(\text{fcl})}(\mathbf{z}_l) \quad (3.1.3)$$

と表記する。このとき、 $L$  個の隠れ層を持つ多層ペーセプトロンは

$$\mathbf{y}^{(\text{FNN})} = \mathbf{h}_{L+1}^{(\text{fcl})} \circ \mathbf{h}_L^{(\text{fcl})} \circ \dots \circ \mathbf{h}_1^{(\text{fcl})} \left( \mathbf{x}^{(\text{FNN})} \right) \quad (3.1.4)$$

と定義される。ここで、 $\circ$  は関数の合成を表す演算子、 $\mathbf{y}^{(\text{FNN})}$  は出力ベクトル、 $\mathbf{x}^{(\text{FNN})}$  は入力ベクトルを表す。

### 3.1.1.2 不動点を含む全結合層

通常、船舶は外力や船速が存在しない場合、加速度は発生しない。つまり、船舶の操縦モデルは状態空間の原点に不動点をもつ不動点方程式と考えられる。本研究では、操縦モデルを表現するために FNN を用いるが、この現象を効率的に表現するための一つの手法として、入力変数がゼロならば出力変数がゼロとなる、中西ら [121] が導入した不動点を含む全結合層を利用することが可能である。不動点を含む全結合層は

$$\mathbf{z}_{l+1} = \mathbf{g}_l(\mathbf{W}_l \mathbf{z}_l + \mathbf{b}_l) - \mathbf{g}_l(\mathbf{b}_l) \quad (3.1.5)$$

と定義される。つまり、どのような重み行列  $\mathbf{W}_l$  とバイアスベクトル  $\mathbf{b}_l$  であったとしても、入力変数の原点は出力変数の原点へ変換される。本論文では、この不動点を含む全結合層の処理は

$$\mathbf{z}_{l+1} = \mathbf{h}_l^{(\text{fpl})}(\mathbf{z}_l) \quad (3.1.6)$$

と表現される。よって、不動点を含む全結合層のみで構成される FNN を用いて操縦モデルを表現することにより、任意のパラメータに対して入力変数がゼロならば出力変数がゼロとなる操縦モデルを表現することが可能である。ただし、一つでも標準の全結合層が含まれてしまうと、この特性は失われてしまうことに注意する。このとき、 $L$  個の不動点を含む全結合層を持つ FNN は

$$\mathbf{y}^{(\text{FNN})} = \mathbf{h}_{L+1}^{(\text{fpl})} \circ \mathbf{h}_L^{(\text{fpl})} \circ \dots \circ \mathbf{h}_1^{(\text{fpl})} \left( \mathbf{x}^{(\text{FNN})} \right) \quad (3.1.7)$$

と定義される。

### 3.1.2 回帰型ニューラルネットワーク

回帰型ニューラルネットワーク (Recurrent Neural Network: RNN) は内部に循環を持つ ANN で, FNN と同様に機械学習タスクで頻繁に使用される代表的な ANN モデルの一つである。RNN は内部に循環を持つことで内部状態をネットワーク内に保持している。そのため, RNN はシーケンスデータにおける時間的な動的特性を表現することが可能である。

RNN は FNN と同様に複数の層によって構成されるが, RNN では前時間ステップで得られた出力結果を再度入力にとる層で構成される。つまり, ある時刻ステップ  $k$  における  $l$  層目の潜在変数を  $z_{k,l} \in \mathbb{R}^{N_{z_l}}$  と表記すると,  $l$  層目から  $l+1$  層目への変換は

$$z_{k,l+1} = g_l (W_l z_{k,l} + W'_l z_{k-1,l+1} + b_l) \quad (3.1.8)$$

と定義される。ここで,  $W_l \in \mathbb{R}^{N_{l+1} \times N_l}$  および  $W'_l \in \mathbb{R}^{N_{l+1} \times N_{l+1}}$  は重み行列,  $b_l \in \mathbb{R}^{N_{l+1}}$  はバイアスペクトルを表す。本論文では, この RNN 層の処理を

$$z_{k,l+1} = h_l^{(rl)} (z_{k,l}, z_{k-1,l+1}) \quad (3.1.9)$$

と表現する。これにより, ある時刻ステップの入力に基づいて計算された中間層の演算結果を内部状態として次のステップへ引き継ぐことが可能となる。そのため, RNN は過去の入力履歴に基づく出力を表現することが可能である。

また, 3.1.1.2 節で不動点を含む全結合層について述べたが, Eq. (3.1.8) においても不動点を取り入れることが可能である。不動点を含む RNN 層は

$$z_{k,l+1} = g_l (W_l z_{k,l} + W'_l z_{k-1,l+1} + b_l) - g_l (b_l) \quad (3.1.10)$$

と表現できる。本論文では, この不動点を含む RNN 層の処理を

$$z_{k,l+1} = h_l^{(frl)} (z_{k,l}, z_{k-1,l+1}) \quad (3.1.11)$$

と表現する。

船舶の操縦運動は, 船体の運動学的変数ベクトル  $\mathbf{x}$ , アクチュエータ状態変数ベクトル  $\mathbf{u}$ , 真風速風向ベクトル  $\mathbf{w}_T$  などの観測可能な変数によって, 必ずしも表現可能であるとは限らない。例えば, 斜航角が大きな運動では渦によって流体力が変化する可能性があるが, 渦は観測することは難しい。しかし, このような部分観測問題では, 過去の履歴を考慮することによって, いくらか改善できる可能性がある。そのため, RNN を用いた操縦モデル推定に関する研究がこれまで多く実施されている。

しかし, 先行研究において操縦モデルのために用いられてきた RNN は, 大きく無限インパルス応答 (Finite Impulse Response: FIR) 型と有限インパルス応答 (Infinite Impulse Response: IIR) 型の二つに分けることができる。前者は過去の全ての入力を通して保持された内部状態に基づいて出力される標準的な RNN で, 後者はある一定ステップ数の過去の入力を通して保持された内部状態に基づいて出力される NARX モデルと似た構造を持つ RNN である。

### 3.1.2.1 無限インパルス応答型 RNN

まず、無限インパルス応答型 RNN について詳述する。ある時刻ステップ  $k$  における  $L$  個の隠れ層を持つ無限インパルス応答型 RNN は

$$\left\{ \begin{array}{l} z_{k,1} = h_1^{(rl)}(x_k^{(RNN)}, z_{k-1,1}) \\ z_{k,2} = h_2^{(rl)}(z_{k,1}, z_{k-1,2}) \\ \vdots \\ y_k^{(RNN)} = h_L^{(rl)}(z_{k,L-1}, y_{k-1}^{(RNN)}) \end{array} \right. \quad (3.1.12)$$

と定義される。ここで、 $x_k^{(RNN)}$  と  $y_k^{(RNN)}$  はそれぞれ時刻ステップ  $k$  における入力変数と出力変数で、 $z_{k,2}, z_{k,3}, \dots, z_{k,L+1}$  は時刻ステップ  $k$  における内部状態変数である。この RNN モデルは、初期時刻ステップにおいて適切な内部状態変数が与えられると、以降全ての時間ステップにおいて過去の全ての入力変数に基づいた値を出力することが可能である。任意の層は必要に応じて、 $h_l^{(frl)}$  や  $h_l^{(fcl)}$  などに置きかえることも可能である。

### 3.1.2.2 有限インパルス応答型 RNN

次に、有限インパルス応答型 RNN について詳述する。過去  $N_m$  ステップの入力が与えられた、ある時刻ステップ  $k$  における  $L$  個の隠れ層を持つ有限インパルス応答型 RNN は

$$\left\{ \begin{array}{l} z_{k,-N_m+1,1} = h_1^{(fcl)}(x_{k-N_m+1}^{(RNN)}) \\ z_{k,-N_m+2,1} = h_1^{(rl)}(x_{k-N_m+2}^{(RNN)}, z_{k,-N_m+1,1}) \\ \vdots \\ z_{k,0,1} = h_1^{(rl)}(x_k^{(RNN)}, z_{k,1,1}) \\ z_{k,-N_m+1,2} = h_2^{(fcl)}(z_{k,-N_m+1,1}) \\ z_{k,-N_m+2,2} = h_2^{(rl)}(z_{k,-N_m+2,1}, z_{k,-N_m+1,2}) \\ \vdots \\ z_{k,0,2} = h_2^{(rl)}(z_{k,0,1}, z_{k,1,2}) \\ \vdots \\ z_{k,-N_m+1,L+1} = h_L^{(fcl)}(z_{k,-N_m+1,L}) \\ z_{k,-N_m+2,L+1} = h_L^{(rl)}(z_{k,-N_m+2,L}, z_{k,-N_m+1,L+1}) \\ \vdots \\ y_k^{(RNN)} = h_L^{(rl)}(z_{k,0,L}, z_{k,1,L+1}) \end{array} \right. \quad \begin{array}{l} \text{first layer process} \\ \text{second layer process} \\ \vdots \\ \text{last layer process} \end{array} \quad (3.1.13)$$

と定義される。ここで、 $x_{k-N_m+1}^{(RNN)}, x_{k-N_m+2}^{(RNN)}, x_k^{(RNN)}$  は時刻ステップ  $k$  から過去  $N_m$  ステップまでの入力変数、 $y_k^{(RNN)}$  は時刻ステップ  $k$  における出力変数、 $z_{k,k',1}, z_{k,k',2}, \dots, z_{k,k',L+1}$  は時刻ステップ  $k$  において  $k'$  前の内部状態変数を表す。ここで、内部状態変数は時刻ステップ  $k$  が更新されると、すべての内部状態変数が再度計算されることに注意する。従って、この RNN モデルは、過去  $N_m$  ステップの入力変数に基づいた値を出力

することが可能である。任意の層は必要に応じて、 $\mathbf{h}_l^{(\text{frl})}$  や  $\mathbf{h}_l^{(\text{fcl})}$  などに置きかえることも可能である。

### 3.1.3 入出力変数の前後処理

3.1.1 節及び3.1.2 節で述べたの FNN や RNN は Eq. (2.2.4b) の右辺を表現可能である。しかし、FNN や RNN は入力および出力変数の周期性や悪スケール性により影響を受けることが知られている。本節では、それらの影響を軽減するため FNN や RNN の入力および出力変数に施した処理について詳述する。

#### 3.1.3.1 周期変数の除去

Eq. (2.2.4b) の右辺を表す操縦モデルは  $\mathbf{N}_{N_m}^{(-)}, \mathbf{U}_{N_m}^{(-)}, \mathbf{W}_{A,N_m}^{(-)}$  を入力にとるが、これらの中に周期変数である相対風向  $\gamma_A$  が含まれる。ある相対風向  $\gamma_A$  に対して、 $2\pi$  を加えた相対風向  $\gamma_A + 2\pi$  は同じ状態を表しているが、ANN からは異なる入力変数として扱われてしまう。そのため、ANN の入力変数から周期変数の除去を行い、相対風速風向ベクトル  $\mathbf{w}_A$  の代わりに船体固定座標系における相対風状態ベクトル  $\mathbf{w}'_A$  を ANN の入力変数として使用する。つまり、現在時刻を含む過去  $N_m$  ステップ間の離散時刻における相対風状態ベクトル  $\mathbf{w}'_A$  をまとめたベクトルを

$$\mathbf{W}'_{A,N_m}^{(-)}(t) \equiv \left( \mathbf{w}'_A^\top(t), \mathbf{w}'_A^\top(t - \Delta t), \dots, \mathbf{w}'_A^\top(t - (N_m - 1)\Delta t) \right)^\top \in \mathbb{R}^{2N_m} \quad (3.1.14)$$

と定義すると、与えられた入力変数  $\mathbf{N}_{N_m}^{(-)}, \mathbf{U}_{N_m}^{(-)}, \mathbf{W}_{A,N_m}^{(-)}$  のうち、 $\mathbf{W}_{A,N_m}^{(-)}$  は  $\mathbf{W}'_{A,N_m}^{(-)}$  に置き換えられる。

#### 3.1.3.2 標準化

入力変数の各成分間でスケール差によって ANN の性能低下を引き起こす可能性がある。そのため、ANN の入力変数に前処理として訓練データ集合の平均及び標準偏差を用いて標準化を実施する。つまり、入力変数ベクトルの中のある一つの変数を  $x_{\text{ANN}}$ 、標準化された変数を  $\bar{x}_{\text{ANN}}$  と表記すると、標準化された変数は

$$\bar{x}_{\text{ANN}} = \frac{x_{\text{ANN}} - \mu_{x_{\text{ANN}}}}{\sigma_{x_{\text{ANN}}}} \quad (3.1.15)$$

と定義される。ただし、 $\mu_{x_{\text{ANN}}}$  と  $\sigma_{x_{\text{ANN}}}$  はそれぞれ変数  $x_{\text{ANN}}$  の訓練データ集合における平均と標準偏差を表す。従って、Eq. (3.1.15) によって標準化された船体の速度変数ベクトル  $\mathbf{v}$ 、アクチュエータ状態変数ベクトル  $\mathbf{u}$ 、相対風状態ベクトル  $\mathbf{w}'_A$  をそれぞれ  $\bar{\mathbf{v}}$ 、 $\bar{\mathbf{u}}$ 、 $\bar{\mathbf{w}}'_A$  と表記すると、ANN の入力変数は

$$\begin{cases} \bar{\mathbf{N}}_{N_m}^{(-)}(t) \equiv \left( \bar{\mathbf{v}}^\top(t), \bar{\mathbf{v}}^\top(t - \Delta t), \dots, \bar{\mathbf{v}}^\top(t - (N_m - 1)\Delta t) \right)^\top \in \mathbb{R}^{3N_m} \\ \bar{\mathbf{U}}_{N_m}^{(-)}(t) \equiv \left( \bar{\mathbf{u}}^\top(t), \bar{\mathbf{u}}^\top(t - \Delta t), \dots, \bar{\mathbf{u}}^\top(t - (N_m - 1)\Delta t) \right)^\top \in \mathbb{R}^{N_u N_m} \\ \bar{\mathbf{W}}'_{A,N_m}^{(-)}(t) \equiv \left( \bar{\mathbf{w}}'_A^\top(t), \bar{\mathbf{w}}'_A^\top(t - \Delta t), \dots, \bar{\mathbf{w}}'_A^\top(t - (N_m - 1)\Delta t) \right)^\top \in \mathbb{R}^{2N_m} \end{cases} \quad (3.1.16)$$

と定義される。

また、出力変数の各成分間におけるスケール差によっても ANN の性能低下を引き起こす可能性がある。そこで、ANN の出力変数に訓練データ集合の平均と標準偏差に基づく逆標準化を実施する。これにより、操縦モデルの出力変数である船体の加速度変数ベクトル  $\dot{\mathbf{v}}$  の各成分間でスケール差が大きかったとしても、ANN の出力変数がとり得る値のスケール差を軽減することが可能である。具体的には、出力変数ベクトルの中のある一つの変数を  $y_{\text{ANN}}$ 、操縦モデルの出力変数ベクトルの中のある一つの変数を  $\dot{y}$  と表記すると、操縦モデル

の出力変数は

$$\dot{\nu} = y_{\text{ANN}} \cdot \sigma_{\dot{\nu}}^{(\text{train})} + \mu_{\dot{\nu}}^{(\text{train})} \quad (3.1.17)$$

と定義される。ただし、 $\mu_{\dot{\nu}}^{(\text{train})}$  と  $\sigma_{\dot{\nu}}^{(\text{train})}$  は  $\dot{\nu}$  の平均と標準偏差を表す。 $\dot{\nu}$  が直接計測されない場合、 $\nu$  の数値時間微分により算出されることに注意する。

### 3.1.4 操縦モデルの表記方法

本研究では、3.1.1節及び3.1.2節で述べたANNモデルと3.1.3節で述べた入出力変数の前後処理によってEq.(2.2.4b)で表される操縦モデルを表現する。以降では、ANNにより表された操縦モデルは

$$\dot{\nu} = \mathbf{F}_{\theta} \left( \mathbf{N}_{N_m}^{(-)}, \mathbf{U}_{N_m}^{(-)}, \mathbf{W}_{A,N_m}^{(-)} \right) \quad (3.1.18)$$

と表記する。ここで、 $\theta$  はFNNもしくはRNNに含まれるすべての層の重み行列とバイアスベクトルをまとめたパラメータベクトルである。また、Eq.(3.1.18)に基づくEq.(2.2.6)やEq.(2.2.7)はそれぞれ

$$\dot{\nu} = \mathbf{F}'_{\theta} \left( \mathbf{X}_{N_m}^{(-)}, \mathbf{U}_{N_m}^{(-)}, \mathbf{W}_{T,N_m}^{(-)} \right), \quad (3.1.19)$$

$$\dot{x} = \mathbf{f}_{\theta} \left( \mathbf{X}_{N_m}^{(-)}, \mathbf{U}_{N_m}^{(-)}, \mathbf{W}_{T,N_m}^{(-)} \right) \quad (3.1.20)$$

と表記する。ただし、Eqs.(3.1.18)–(3.1.20)がマルコフ特性を持つと仮定する場合は、 $N_m = 0$ となり  $\mathbf{X}_{N_m}^{(-)}$ ,  $\mathbf{H}_{N_m}^{(-)}$ ,  $\mathbf{N}_{N_m}^{(-)}$ ,  $\mathbf{U}_{N_m}^{(-)}$ ,  $\mathbf{W}_{T,N_m}^{(-)}$ ,  $\mathbf{W}_{A,N_m}^{(-)}$  はすべて  $x$ ,  $\eta$ ,  $\nu$ ,  $u$ ,  $w_A$ ,  $w_T$  に置き換わることに注意する。

## 3.2 操縦モデルの推定手法

本節では、3.1節で述べた操縦モデルに含まれるパラメータの推定手法について詳述する。一般に、SIでは、計測データに対してモデルがどの程度実際の応答結果を表現できているかを表す損失関数が定義され、そして、その損失関数を最適化することによって計測データを十分に説明可能なモデルが作成される。

ANNを用いた船舶の操縦モデル推定に関するいくつかの研究[25, 38, 42–44]では、回帰分析の形式論に還元されることで操縦モデルが同定されている。つまり、対象システムの入出力変数の計測データを用いて操縦モデルのパラメータが決定されている。しかし、このような同定手法は加速度の情報が必要不可欠である。Sutuloら[20, 23]が言及しているように、加速度の記録は加速度計や速度の数値的な導出によって取得することも可能であるが、加速度記録は外乱に敏感で、通常、ノイズで大きく汚染されている。

そこで、本研究では、加速度の計測値を必要とせず、計測が容易な船体の運動学的変数ベクトル  $\mathbf{x}$  と操縦運動に影響を与えるその他の入力変数に関する計測値のみを用いたANNを用いた船舶の操縦モデル推定手法を提案する。この手法では、船体の運動学的変数ベクトル  $\mathbf{x}$  に関する計測軌道と操縦モデルを用いてシミュレートされた推定軌道の誤差を最小化することでパラメータが同定される。

以降、3.2.1節で回帰分析ベースの手法、3.2.2節で軌道推定ベースの手法について述べる。その後、3.2.3節でANNパラメータの最適化手法について述べる。

### 3.2.1 回帰分析ベースの手法

まず、回帰分析問題を解くことにより操縦モデルの同定を行う手法について詳述する。この手法では、操縦モデルの入出力変数、すなわち、船体の速度変数ベクトル  $\mathbf{v}$ 、船体の加速度変数ベクトル  $\dot{\mathbf{v}}$ 、アクチュエータ

状態変数ベクトル  $\mathbf{u}$  および相対風速風向ベクトル  $\mathbf{w}_A$  に関する計測値が要求される。ここでは、 $N$  個の計測データの組み合わせが与えられると仮定され、データ集合は

$$\mathcal{D}^{(\text{reg})} = \left\{ \left( \dot{\mathbf{v}}_n, \mathbf{N}_{N_m, n}^{(-)}, \mathbf{U}_{N_m, n}^{(-)}, \mathbf{W}_{A, N_m, n}^{(-)} \right) \right\}_{n=1, \dots, N} \quad (3.2.1)$$

と定義される。ここで、下付き文字  $n$  の順序は時間的な制約を持たないが、 $\mathbf{N}_{N_m, n}^{(-)}$  に含まれる変数の順序には時間的な制約を持つことに注意する。

このとき、操縦モデルの損失関数は

$$\mathcal{L}^{(\text{reg})} (\boldsymbol{\theta}; \mathcal{D}^{(\text{reg})}) = \frac{1}{N} \sum_{n=1}^N \left\| \dot{\mathbf{v}}_n - \mathbf{F}_{\boldsymbol{\theta}} \left( \mathbf{N}_{N_m, n}^{(-)}, \mathbf{U}_{N_m, n}^{(-)}, \mathbf{W}_{A, N_m, n}^{(-)} \right) \right\|_{\mathbf{W}_{\dot{\mathbf{v}}}}^2 \quad (3.2.2)$$

と定義される。ここで、 $\mathbf{W}_{\dot{\mathbf{v}}} \in \mathbb{R}^{3 \times 3}$  は重み行列を表すハイパーパラメータである。この重み行列  $\mathbf{W}_{\dot{\mathbf{v}}}$  の選択手法にはいくつかの候補が考えられ、適切に選択される必要がある。

- 一つ目は最も簡単に単位行列を選択する方法である。この場合、損失関数は単なる二乗誤差の形となる。しかし、船体の加速度変数ベクトル  $\dot{\mathbf{v}}$  の要素はそれぞれ単位が異なり、また、分布している範囲も異なる可能性がある。そのため、単位行列を選択すると、単位の選び方やデータ集合の分布によって大きく結果が変化する可能性がある。
- 二つ目の方法はそれぞれの変数の分散に応じた重み付けを行う方法である。この方法では、船体の加速度変数ベクトル  $\dot{\mathbf{v}}$  の成分毎の分散の逆数を対角成分を持つ行列を選択される。つまり、与えられたデータ集合  $\mathcal{D}^{(\text{reg})}$  における船体の加速度変数ベクトル  $\dot{\mathbf{v}}$  の成分毎の標準偏差をそれぞれ  $\sigma_{\dot{u}}^{(\text{train})}, \sigma_{\dot{v}_m}^{(\text{train})}, \sigma_{\dot{r}}^{(\text{train})}$  と表記すると、重み行列は  $\mathbf{W}_{\dot{\mathbf{v}}} = \text{diag} \left( \left( \sigma_{\dot{u}}^{(\text{train})} \right)^{-2}, \left( \sigma_{\dot{v}_m}^{(\text{train})} \right)^{-2}, \left( \sigma_{\dot{r}}^{(\text{train})} \right)^{-2} \right)$  と選択される。この場合、損失関数は標準化された船体の加速度変数ベクトル  $\dot{\mathbf{v}}$  に関する計測値と推定値の二乗誤差に相当する。3.3 節で述べられる検証実験ではこの方法が用いられた。
- 三つ目の方法はそれぞれの変数の観測誤差の大きさに応じた重み付けを行う方法である。船体の加速度変数ベクトル  $\dot{\mathbf{v}}$  の成分毎の観測誤差がゼロ平均ガウス分布に従うと仮定し、船体の加速度変数ベクトル  $\dot{\mathbf{v}}$  の成分毎の標準偏差をそれぞれ  $\sigma_{\dot{u}}^{(\text{obs})}, \sigma_{\dot{v}_m}^{(\text{obs})}, \sigma_{\dot{r}}^{(\text{obs})}$  と表記すると重み行列は  $\mathbf{W}_{\dot{\mathbf{v}}} = \text{diag} \left( \left( \sigma_{\dot{u}}^{(\text{obs})} \right)^{-2}, \left( \sigma_{\dot{v}_m}^{(\text{obs})} \right)^{-2}, \left( \sigma_{\dot{r}}^{(\text{obs})} \right)^{-2} \right)$  と選択される。この場合、損失関数は最尤推定における負の対数尤度に相当する。

### 3.2.2 軌道推定ベースの手法

次に、船体の運動学的変数ベクトル  $\mathbf{x}$  に関する計測軌道と操縦モデルを用いてシミュレートされた推定軌道の誤差を最小化することで操縦モデルの同定を行う手法について詳述する。この手法では、船舶の操縦運動における船体の運動学的変数ベクトル  $\mathbf{x}$ 、アクチュエータ状態変数ベクトル  $\mathbf{u}$ 、真風速風向ベクトル  $\mathbf{w}_T$  に関する観測値の時系列データが要求される。ここで、 $N$  個の時系列データが与えられると仮定し、 $n$  番目の時系列データはある時刻  $t_n$  を初期時刻とし、 $\Delta t$  の時間間隔を持つ  $K$  個の時刻点  $\{t_n + k\Delta t\}_{k=0}^{K-1}$  で計測されたものと

する。まず、 $\mathbf{x}$ ,  $\mathbf{u}$  および  $\mathbf{w}_T$  に関する  $n$  番目の計測変数列をそれぞれ

$$\begin{cases} \mathbf{X}_{K,n}^{(+)} = (\mathbf{x}^\top(t_n), \mathbf{x}^\top(t_n + \Delta t), \dots, \mathbf{x}^\top(t_n + (K-1)\Delta t))^\top \in \mathbb{R}^{6K} \\ \mathbf{U}_{K,n}^{(+)} = (\mathbf{u}^\top(t_n), \mathbf{u}^\top(t_n + \Delta t), \dots, \mathbf{u}^\top(t_n + (K-1)\Delta t))^\top \in \mathbb{R}^{N_u K} \\ \mathbf{W}_{T,K,n}^{(+)} = (\mathbf{w}_T^\top(t_n), \mathbf{w}_T^\top(t_n + \Delta t), \dots, \mathbf{w}_T^\top(t_n + (K-1)\Delta t))^\top \in \mathbb{R}^{2K} \end{cases} \quad (3.2.3)$$

と定義する。このとき、与えられるデータ集合は

$$\mathcal{D}^{(\text{traj})} = \left\{ \left( \mathbf{X}_{K,n}^{(+)}, \mathbf{U}_{K,n}^{(+)}, \mathbf{W}_{T,K,n}^{(+)} \right) \right\}_{n=1,\dots,N} \quad (3.2.4)$$

と定義される。以降では、このデータ集合  $\mathcal{D}^{(\text{traj})}$  を用いて操縦運動軌道が行われる。

### 3.2.2.1 操縦運動軌道の推定

ここでは、入力変数列  $\mathbf{U}_{K,n}^{(+)}, \mathbf{W}_{T,K,n}^{(+)}$  から運動学的変数列  $\mathbf{X}_{K,n}^{(+)}$  を操縦モデルによって推定する問題を考える。操縦モデルによって推定された運動学的変数を  $\mathbf{x}_\theta$  とし、運動学的変数列を

$$\mathbf{X}_{\theta,K,n}^{(+)} = (\mathbf{x}_\theta^\top(t_n), \mathbf{x}_\theta^\top(t_n + \Delta t), \dots, \mathbf{x}_\theta^\top(t_n + (K-1)\Delta t))^\top \in \mathbb{R}^{6K} \quad (3.2.5)$$

と表記する。以降において、この運動学的変数  $\mathbf{x}_\theta$  の推定手法について詳述する。

#### (1) 操縦モデルがマルコフ特性を持つ場合

ある時刻  $t$  における運動学的変数  $\mathbf{x}_\theta$  は

$$\mathbf{x}_\theta(t) = \mathbf{x}_\theta(t_n) + \int_{\tau=t_n}^t \mathbf{f}_\theta(\mathbf{x}_\theta(\tau), \mathbf{u}(\tau), \mathbf{w}_T(\tau)) d\tau \quad \text{for } t_n \leq t \leq t_n + (K-1)\Delta t \quad (3.2.6)$$

を解くことによって求められる。ただし、 $\mathbf{x}_\theta(t_n)$  は与えられる必要があり、通常、計測値が代入される、つまり  $\mathbf{x}_\theta(t_n) = \mathbf{x}(t_n)$  とされることが多い。しかし、操縦モデルパラメータと同時に最適化されるパラメータとして扱うことも可能であることに注意する。また、Eq. (3.2.6) の積分は 4 次の Runge–Kutta 法や Euler 法などを用いて数値的に解くことが可能であるが、本研究では Euler 法を用いる。つまり、運動学的変数  $\mathbf{x}_\theta$  は

$$\mathbf{x}_\theta(t_n + k\Delta t) = \mathbf{x}(t_n) + \sum_{i=0}^{k-1} \Delta t \cdot \mathbf{f}_\theta(\mathbf{x}_\theta(t_n + i\Delta t), \mathbf{u}(t_n + i\Delta t), \mathbf{w}_T(t_n + i\Delta t)) \quad (3.2.7)$$

を解くことによって求められる。

#### (2) 操縦モデルが過去の有限ステップの履歴を考慮している場合

ある時刻  $t$  における運動学的変数  $\mathbf{x}_\theta$  は

$$\mathbf{x}_\theta(t_n + k\Delta t) = \begin{cases} \mathbf{x}(t_n + k\Delta t) & \text{for } k < N_m \\ \mathbf{x}(t_n + (N_m - 1)\Delta t) + \\ \sum_{i=M-1}^{k-1} \Delta t \cdot \mathbf{f}_\theta \left( \mathbf{X}_{N_m,\theta}^{(-)}(t_n + i\Delta t), \mathbf{U}_{N_m}^{(-)}(t_n + i\Delta t), \mathbf{W}_{T,N_m}^{(-)}(t_n + i\Delta t) \right) & \text{for } N_m \leq k \end{cases} \quad (3.2.8)$$

と表される。ここで、 $\mathbf{x}(t_n), \mathbf{x}(t_n + \Delta t), \dots, \mathbf{x}(t_n + (N_m - 1)\Delta t)$  をパラメータとして置き換えることも可能であるが、本研究では、パラメータ数が多くなるためこの手法では計測値が直接代入されることに注意する。

(3) 操縦モデルが過去の全ての履歴を考慮している場合

ある時刻  $t$  における運動学的変数  $\mathbf{x}_\theta$  は

$$\mathbf{x}_\theta(t_n + k\Delta t) = \mathbf{x}(t_n) + \sum_{i=0}^{k-1} \Delta t \cdot \mathbf{f}_\theta \left( X_{i,\theta}^{(-)}(t_n + i\Delta t), U_i^{(-)}(t_n + i\Delta t), W_{T,i}^{(-)}(t_n + i\Delta t) \right) \quad (3.2.9)$$

と表される。ここで、履歴ステップ数  $N_m$  が進行した時間ステップ数  $i$  に応じて変化しており、過去全ての履歴が操縦モデルに入力されていることに注意する。また、この手法でも  $\mathbf{x}(t_n)$  をパラメータとして置き換えることも可能であるが、本研究では、計測値が直接代入されることに注意する。

### 3.2.2.2 損失関数

運動学的変数の計測軌道と操縦モデルを用いてシミュレートされた運動学的変数の推定軌道の誤差の大きさを表す損失関数について詳述する。操縦モデルの損失関数は

$$\mathcal{L}^{(\text{traj})}(\boldsymbol{\theta}; \mathcal{D}^{(\text{traj})}) = \frac{1}{NK} \sum_{n=1}^N \sum_{k=1}^K \|\mathbf{x}(t_n + k\Delta t) - \mathbf{x}_\theta(t_n + k\Delta t)\|_{W_x}^2 \quad (3.2.10)$$

と定義される。ここで、 $W_x \in \mathbb{R}^{6 \times 6}$  は重み行列を表すハイパーパラメータである。この重み行列  $W_x$  の選択手法は Eq. (3.2.2) の重み行列と同様に以下の三つの候補が考えられる。

- 一つ目は単位行列を選択する方法である。この場合、3.2.1 節でも述べたが、要素毎の単位の選び方やデータ集合の分布によって大きく結果が変化する可能性があることに注意したい。
- 二つ目の方法はそれぞれの変数の分散に応じた重み付けを行う方法である。この方法では、船体の運動学的変数ベクトル  $\mathbf{x}$  の成分毎の分散の逆数を対角成分を持つ行列を選択される。つまり、与えられたデータ集合  $\mathcal{D}^{(\text{traj})}$  における船体の運動学的変数ベクトル  $\mathbf{x}$  の成分毎の標準偏差をそれぞれ  $\sigma_{x_0}^{(\text{train})}, \sigma_{y_0}^{(\text{train})}, \sigma_\psi^{(\text{train})}, \sigma_u^{(\text{train})}, \sigma_{v_m}^{(\text{train})}, \sigma_r^{(\text{train})}$  と表記すると重み行列は  $W_x = \text{diag} \left( \left( \sigma_{x_0}^{(\text{train})} \right)^{-2}, \left( \sigma_{y_0}^{(\text{train})} \right)^{-2}, \left( \sigma_\psi^{(\text{train})} \right)^{-2}, \left( \sigma_u^{(\text{train})} \right)^{-2}, \left( \sigma_{v_m}^{(\text{train})} \right)^{-2}, \left( \sigma_r^{(\text{train})} \right)^{-2} \right)$  と選択される。この場合、損失関数は標準化された船体の運動学的変数ベクトル  $\mathbf{x}$  に関する計測値と推定値の二乗誤差に相当する。3.3 節で述べられる検証実験ではこの方法が用いられた。
- 三つ目の方法はそれぞれの変数の観測誤差の大きさに応じた重み付けを行う方法である。船体の運動学的変数ベクトル  $\mathbf{x}$  の成分毎の観測誤差がゼロ平均ガウス分布に従うと仮定し、船体の運動学的変数ベクトル  $\mathbf{x}$  の成分毎の標準偏差をそれぞれ  $\sigma_{x_0}^{(\text{obs})}, \sigma_{y_0}^{(\text{obs})}, \sigma_\psi^{(\text{obs})}, \sigma_u^{(\text{obs})}, \sigma_{v_m}^{(\text{obs})}, \sigma_r^{(\text{obs})}$  と表記すると、重み行列は  $W_x = \text{diag} \left( \left( \sigma_{x_0}^{(\text{obs})} \right)^{-2}, \left( \sigma_{y_0}^{(\text{obs})} \right)^{-2}, \left( \sigma_\psi^{(\text{obs})} \right)^{-2}, \left( \sigma_u^{(\text{obs})} \right)^{-2}, \left( \sigma_{v_m}^{(\text{obs})} \right)^{-2}, \left( \sigma_r^{(\text{obs})} \right)^{-2} \right)$  と選択される。この場合、損失関数は最尤推定における負の対数尤度に相当する。4.2 節で述べられる検証実験ではこの方法が用いられた。

### 3.2.3 ANN パラメータの最適化手法

本研究では、3.2.1 節及び3.2.2 節で述べた損失関数を最小化することで操縦モデルの推定を行う。この最小化のため、データ集合を小さなバッチに分割して勾配を計算するミニバッチ学習と勾配降下法ベースの最適化手法である Adam [122] が用いられる。ただし、過学習回避のため、与えられたデータ集合は訓練データ集合

と評価データ集合の二つに分けられる。そして訓練データ集合は最適化のために使用され、評価データ集合はオーバーフィッティングの開始を検出するために使用される。つまり、訓練が終了すると、エポック毎に記録されたパラメータの中から、評価データ集合に対する損失関数が最も小さくなるパラメータを最適なパラメータ  $\theta^*$  とした。つまり、過学習が発生する前のパラメータを採用することで過学習を回避している。本研究では、ANN モデルの構築および訓練の実装には、Python の機械学習ライブラリの一つである PyTorch が使用された。

### 3.3 模型試験データを用いた検証実験

本節では、3.1 節で述べた RNN モデルと3.2 節で述べた訓練手法を用いた操縦モデルの推定実験の結果を示す。ここで示す内容は、自著論文 1 [28] で示された内容と同様である。この実験では、2.3 節で述べた供試船 1 が対象船舶とされている。以降、3.3.1 節で使用された操縦運動データ集合について詳述し、3.3.2 節で実施した実験内容について、3.3.3 節でその結果について示す。

#### 3.3.1 操縦運動データ集合

本実験で用いられる操縦運動データは2.5 節で述べた大阪大学の実験池である犬飼池にて実施された供試船 1 の自由航走試験により収集された。自由航走試験の詳細を以降に述べる。

操縦モデルのシステム同定に関する多くの研究では、主に、十分大きく安定した船速を持つ標準的な操船運動を推定することが目的とされている。そのため、多くの研究はジグザグ操船試験と旋回試験で計測された操縦運動データを採用している。一方で、本研究の目的は標準的な操船運動に加えて低速操縦運動を推定可能な操縦モデルを獲得することであり、操縦モデルは斜航角の大きな運動や後進運動もしくはプロペラ逆転などの状態に対する操縦運動を適切に推定することが求められる。そこで、本研究では、低速操縦運動でとり得る様々な運動状態を計測するため、ジグザグ操船試験と旋回試験に加えて、ランダム操船試験と仮想岸壁を考慮した手動着桟操船試験が実施された。

ランダム操船試験は、船舶がとり得る船体の運動学的変数ベクトル  $\mathbf{x}$  およびアクチュエータ状態変数ベクトル  $\mathbf{u}$  に関するデータを効率的に収集することを目的に、アクチュエータ状態変数ベクトル  $\mathbf{u}$  がランダムに決定された自由航走試験である。本実験では、時々刻々のアクチュエータ状態変数ベクトル  $\mathbf{u}$  はゲームコントローラを通して人間により決定された。ランダム操船試験の操縦運動軌道を Fig. 3.1 に示し、ランダム操船試験で入力されたアクチュエータ状態変数ベクトル  $\mathbf{u}$  のヒストグラムを Fig. 3.2 に示す。Fig. 3.1 を見ると、ランダム操船試験では標準的な操船運動には含まれない、斜航角の大きな運動や後進運動などが計測されていることがわかる。また、Fig. 3.2 を見るとアクチュエータ状態変数ベクトル  $\mathbf{u}$  は一様に分布しているわけではないことが分かる。これは、人間のオペレータが実験池の形状を考慮しつつ、入力を決定したためである。しかし、偏りはあるものの幅広いアクチュエータ状態変数ベクトル  $\mathbf{u}$  のパターンが入力されていることが分かる。

本実験で用いられるデータは、2.3.1 節で述べた計測システムによって得られる。 $\mathcal{D}^{(\text{reg})}$  は船体の加速度変数ベクトル  $\mathbf{v}$  に関するデータを必要とするが計測システムでは直接計測されない。そのため、船体の速度変数ベクトル  $\mathbf{v}$  の数値時間微分によって得られていることに注意されたい。

実施された自由航走試験の操縦運動を Table 3.1 にまとめる。これらの操縦運動データを用いて訓練および検証のためのデータ集合を用意した。本実験では、Table 3.2 に示す三つの訓練データ集合および一つの検証データ集合を用意した。ここで、Train-TZB および Train-TZRB は同程度のデータ量を持ち、前者は旋回操

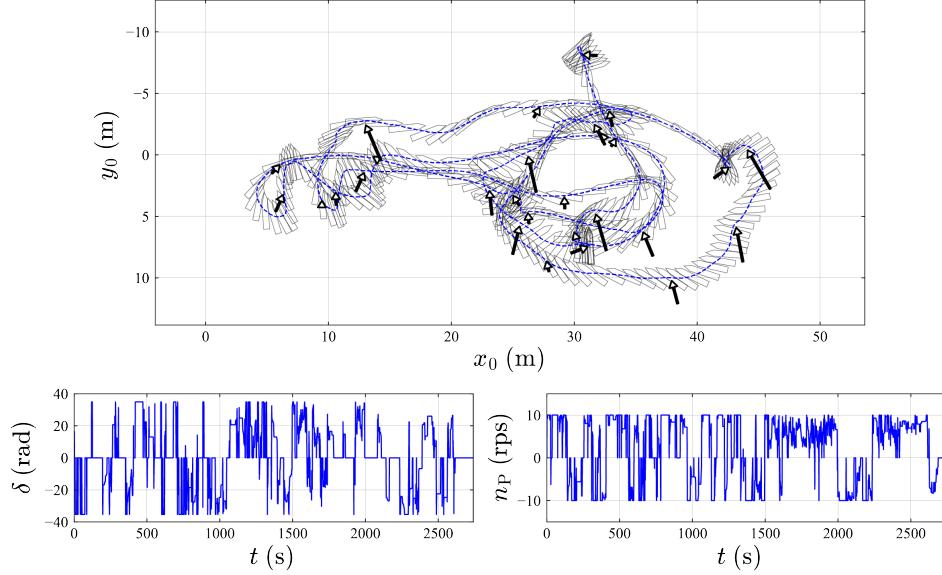


Fig. 3.1: An example of maneuvering motions in a random maneuvering test. The arrows indicate the wind speed and direction at that moment.

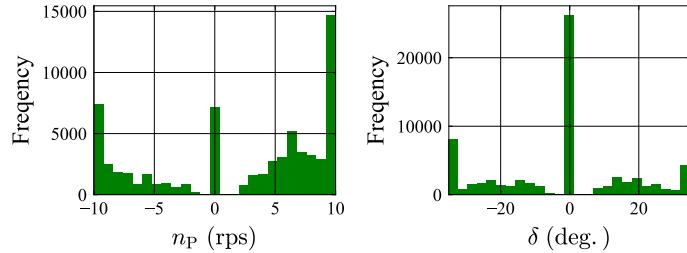


Fig. 3.2: Histogram of actuator state  $\mathbf{u}$  in a random maneuvering test. In this test, propeller revolution  $n_P$  is limited in  $[-10, 10]$ .

船, ジグザグ操船, 着棧操船のみで構成された訓練データ集合で, 後者はランダム操船が含まれている. ここで, Train-TZB および Train-TZRB のデータ量は実船スケール換算では約 7 時間程度となるよう, データ量が制限されている. 一方で, データ量の増減による操縦モデルの推定精度の変化を確認するため, 5 倍程度のデータ量を持つ Train-TZRB+ を用意した.

### 3.3.2 実験内容

実施した操縦モデルの推定実験の詳細な設定について述べる. 3.3.1 節で述べた訓練データ集合を用いて操縦モデルの推定を行い, いくつかの比較検証を実施した. 本実験では無限インパルス応答型 RNN および有限インパルス応答型 RNN のそれぞれを用いた操縦モデルの推定結果の比較を実施した. 実験に用いられた無限

Table 3.1: Notations for maneuvers.

Notation	Description
Z	A zigzag maneuver.
T	A turning maneuver.
R	A random maneuver.
B	A berthing maneuver.

Table 3.2: Training and test datasets and its duration.

Name	T (s)	Z (s)	R (s)	B (s)	Total (s)
Train-TZB	1490.0	737.1	0.0	335.8	2562.9
Train-TZRB	556.4	342.9	1301.2	335.8	2536.3
Train-TZRB+	5674.7	1151.0	5861.2	788.9	13475.8
Test-TZRB	424.6	193.8	717.9	380.6	1716.9

インパルス応答型 RNN は

$$\begin{cases} z_{k,2} = \mathbf{h}_1^{(\text{frl})} (\mathbf{x}_k^{(\text{RNN})}, z_{k-1,2}) \\ z_{k,3} = \mathbf{h}_2^{(\text{fpl})} (z_{k,2}) \\ z_{k,4} = \mathbf{h}_3^{(\text{fpl})} (z_{k,3}) \\ \mathbf{y}_k^{(\text{RNN})} = \mathbf{h}_4^{(\text{fpl})} (z_{k,4}) \end{cases} \quad (3.3.1)$$

と定義され、また、有限インパルス応答型 RNN は

$$\begin{cases} z_{k,-N_m+1,2} = \mathbf{h}_1^{(\text{fpl})} (\mathbf{x}_{k-N_m+1}^{(\text{RNN})}) \\ z_{k,-N_m+2,2} = \mathbf{h}_1^{(\text{frl})} (\mathbf{x}_{k-N_m+2}^{(\text{RNN})}, z_{k,-N_m+1,2}) \\ \vdots \\ z_{k,0,2} = \mathbf{h}_1^{(\text{frl})} (\mathbf{x}_k^{(\text{RNN})}, z_{k,1,2}) \\ z_{k,3} = \mathbf{h}_2^{(\text{fpl})} (z_{k,2}) \\ z_{k,4} = \mathbf{h}_3^{(\text{fpl})} (z_{k,3}) \\ \mathbf{y}_k^{(\text{RNN})} = \mathbf{h}_4^{(\text{fpl})} (z_{k,4}) \end{cases} \quad (3.3.2)$$

と定義される。ここで、 $\mathbf{h}_1^{(\text{frl})}, \mathbf{h}_1^{(\text{fpl})}, \mathbf{h}_2^{(\text{fpl})}, \mathbf{h}_3^{(\text{fpl})}, \mathbf{h}_4^{(\text{fpl})}$  は Table 3.4 で定義される不動点を含む RNN 層および不動点を含む全結合層である。

また、本実験では、3.2.1 節で述べた回帰分析ベースおよび3.2.2 節で述べた軌道推定ベースの操縦モデルの訓練手法の比較が実施される。つまり、Eq. (3.2.2) および Eq. (3.2.10) でそれぞれ定義される損失関数の最小化によって得られた操縦モデルの比較を行う。ただし、損失関数における重み行列  $\mathbf{W}_v$  および  $\mathbf{W}_x$  はそれぞ

Table 3.3: The experiment patterns for estimation of maneuvering model

Name	RNN model	Training method	Dataset
Type-1	FIR-RNN (Eq. (3.3.2))	Trajectory estimation (3.2.2 節)	Train-TZRB+
Type-2	IIR-RNN (Eq. (3.3.1))	Trajectory estimation (3.2.2 節)	Train-TZRB+
Type-3	FIR-RNN (Eq. (3.3.2))	Regression (3.2.1 節)	Train-TZRB+
Type-4	FIR-RNN (Eq. (3.3.2))	Trajectory estimation (3.2.2 節)	Train-TZB
Type-5	FIR-RNN (Eq. (3.3.2))	Trajectory estimation (3.2.2 節)	Train-TZRB

れの変数成分の訓練データ集合における分散に応じた重み付け方法によって決定される。つまり、

$$\begin{aligned} \mathbf{W}_v &= \text{diag} \left( \left( \sigma_{\dot{u}}^{(\text{train})} \right)^{-2}, \left( \sigma_{\dot{v}_m}^{(\text{train})} \right)^{-2}, \left( \sigma_{\dot{r}}^{(\text{train})} \right)^{-2} \right), \\ \mathbf{W}_x &= \text{diag} \left( \left( \sigma_{x_0}^{(\text{train})} \right)^{-2}, \left( \sigma_{y_0}^{(\text{train})} \right)^{-2}, \left( \sigma_{\psi}^{(\text{train})} \right)^{-2}, \left( \sigma_u^{(\text{train})} \right)^{-2}, \left( \sigma_{v_m}^{(\text{train})} \right)^{-2}, \left( \sigma_r^{(\text{train})} \right)^{-2} \right) \end{aligned} \quad (3.3.3)$$

と定義された。ここで、 $\sigma_{x_0}^{(\text{train})}$ ,  $\sigma_{y_0}^{(\text{train})}$ ,  $\sigma_{\psi}^{(\text{train})}$ ,  $\sigma_u^{(\text{train})}$ ,  $\sigma_{v_m}^{(\text{train})}$ ,  $\sigma_r^{(\text{train})}$ ,  $\sigma_{\dot{u}}^{(\text{train})}$ ,  $\sigma_{\dot{v}_m}^{(\text{train})}$ ,  $\sigma_{\dot{r}}^{(\text{train})}$  は成分毎の訓練データ集合における標準偏差であることに注意する。

本実験では、Table 3.3に示す Type-1 から Type-5 までの五通りの設定で実験が行われた。また、実験で用いられたハイパーパラメータを Table 3.5に示す。この実験の目的を以下に示す:

- 3.1.2.1 節で述べた無限インパルス応答型 RNN と3.1.2.2 節で述べた有限インパルス応答型 RNN を用いた予測精度の違いを比較し、履歴影響の長さによる精度の違いを確認する。これは、Type-1 と Type-2 を比較することで確認する。
- 3.2.1 節で述べた回帰分析ベースの操縦モデル推定手法と3.2.2 節で述べた軌道推定ベースの操縦モデル推定手法を用いた予測精度の違いを比較する。これは、Type-1 と Type-3 を比較することで確認する。
- 与えられたデータ集合の違いによる予測精度の違いを比較する。特に、データ量の違いとランダム操船試験の有無による精度の違いを確認する。これは、Type-1, Type-4 および Type-5 を比較することで確認する。
- 3.2.3 節で述べた拘束模型試験および経験式によって係数が決定された MMG モデルと RNN モデルの予測精度の違いを比較し、RNN モデルの有効性を確認する。これは、Type-1 から Type-5 の中で最も良い予測精度を持つ結果を MMG モデルの予測精度と比較することで確認する。

一般に、ANN モデルの訓練結果はパラメータの初期値のような訓練過程に含まれるランダム性に影響を受ける可能性がある。そのため、本研究では全ての実験に対して、異なる五つの初期パラメータに対して同じ訓練を実施した。訓練では、3.2.3 節で述べたように、与えられたデータ集合が訓練データ集合と評価データ集合に分割され、訓練データ集合を用いて損失関数の最小化が行われ、評価データ集合を用いて過学習の検出が行われる。本実験の訓練における評価データ集合に対する損失関数の値を Fig. 3.3に示す。Fig. 3.3から、評価データ集合に対する損失関数の値は訓練過程に含まれるランダム性によって大きく変動していないため、操縦モデルの訓練結果も変動していないことがわかる。

Table 3.4: The used network layers for maneuvering model.

	Dimension of $\mathbf{W}_l$	Dimension of $\mathbf{W}'_l$	Dimension of $\mathbf{b}_l$	Activation function $g$
$\mathbf{h}_1^{(\text{frl})}$	$\mathbb{R}^{200 \times 7}$	$\mathbb{R}^{200 \times 200}$	$\mathbb{R}^{200}$	hyperbolic tangent
$\mathbf{h}_1^{(\text{fp1})}$	$\mathbb{R}^{200 \times 7}$	None	$\mathbb{R}^{200}$	hyperbolic tangent
$\mathbf{h}_2^{(\text{fp1})}$	$\mathbb{R}^{200 \times 200}$	None	$\mathbb{R}^{200}$	hyperbolic tangent
$\mathbf{h}_3^{(\text{fp1})}$	$\mathbb{R}^{200 \times 200}$	None	$\mathbb{R}^{200}$	hyperbolic tangent
$\mathbf{h}_4^{(\text{fp1})}$	$\mathbb{R}^{3 \times 200}$	None	$\mathbb{R}^3$	None

Table 3.5: Hyperparameters in training.

Item	Value
Batch size	512
Learning rate for trajectory estimation method (3.2.2 節)	$2.0 \times 10^{-5}$
Learning rate for regression method (3.2.1 節)	$1.0 \times 10^{-4}$
Duration of time step: $\Delta t$	0.1 (s)
Number of time steps: $K$	60 (6 (s))
Number of memory steps: $N_m$	10 (1 (s))
Number of Epochs	20,000

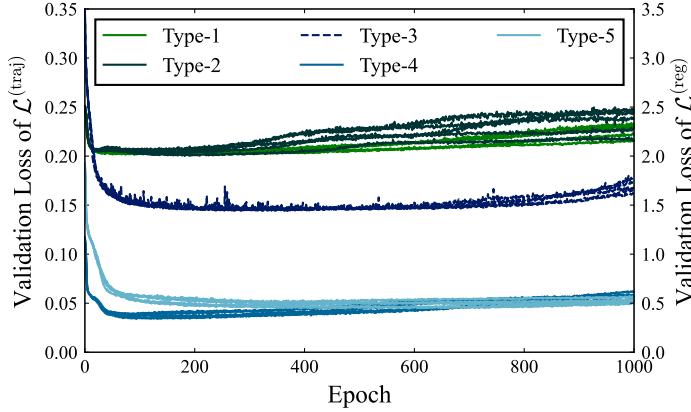


Fig. 3.3: Validation loss for each epoch during training. Note that the Type-3 values are on the secondary vertical axis because they use a different loss function.

### 3.3.3 実験結果

3.3.2 節で訓練された操縦モデルの検証データ集合に対する予測精度を示す。本実験では、得られた全ての操縦モデルの予測精度を確認するため、検証データ集合に含まれるランダム操船、旋回操船、ジグザグ操船、三つの着桟操船に対して、拘束模型試験および経験式によって係数が決定された MMG モデルと訓練された

Table 3.6: Mean and standard deviation of  $\mathcal{L}^{(\text{traj})}$  for a test trajectory (mean  $\pm$  std).

Traj.	EDF-MMG	Type-1	Type-2	Type-3	Type-4	Type-5
R	0.272	$0.236 \pm 0.003$	$0.241 \pm 0.005$	$4834 \pm 4041$	$0.488 \pm 0.022$	$0.262 \pm 0.007$
T	0.858	$0.713 \pm 0.011$	$0.728 \pm 0.017$	$0.864 \pm 0.100$	$0.701 \pm 0.019$	$0.785 \pm 0.024$
Z	1.015	$0.086 \pm 0.063$	$0.124 \pm 0.141$	$761.1 \pm 1519$	$0.122 \pm 0.054$	$0.856 \pm 0.288$
B <sub>1</sub>	0.036	$0.052 \pm 0.024$	$0.058 \pm 0.031$	$634.8 \pm 1264$	$0.144 \pm 0.053$	$0.055 \pm 0.030$
B <sub>2</sub>	0.189	$0.069 \pm 0.014$	$0.140 \pm 0.101$	$91.76 \pm 124.6$	$0.246 \pm 0.107$	$0.182 \pm 0.047$
B <sub>3</sub>	0.100	$0.138 \pm 0.023$	$0.167 \pm 0.053$	$1.163 \pm 0.617$	$0.302 \pm 0.062$	$0.294 \pm 0.043$

RNN モデルを用いて軌道予測を行った。ただし、ランダム操縦および旋回操船は計測時間が長いため、操縦モデルを用いた軌道予測は 100 s 毎に計測値へ初期化された。Eq. (3.3.1) の RNN モデルを用いた軌道予測では初期時刻ステップの入力のみを必要とするが、Eq. (3.3.2) の RNN モデルは初期時刻から  $N_m$  ステップ後までの入力を必要とする。そのため、RNN モデルの推定時間幅を統一するため、Eq. (3.3.1) の RNN モデルは初期時刻から  $N_m$  ステップ後から推定を開始した。

予測結果を定量的に評価するため  $\mathcal{L}^{(\text{traj})}$  の値を求めた。ただし、重み行列  $\mathbf{W}_x$  を構成する  $\sigma_u^{(\text{train})}$ ,  $\sigma_{v_m}^{(\text{train})}$ ,  $\sigma_r^{(\text{train})}$ ,  $\sigma_{\dot{u}}^{(\text{train})}$ ,  $\sigma_{\dot{v}_m}^{(\text{train})}$ ,  $\sigma_r^{(\text{train})}$  は成分毎の検証データ集合における標準偏差が用いられている。得られた結果を Table 3.6 に示す。この結果を基に、予測精度の比較結果を以下に述べる。

### (1) 操縦モデルの比較

まず、操縦モデルに用いられた RNN モデルの比較結果について述べる。Type-1 と Type-2 では、同一の訓練データ集合および訓練手法が用いられたが、使用された操縦モデルがそれぞれ異なる。Type-1 では Eq. (3.3.1) で定義される無限インパルス応答型 RNN が用いられ、Type-2 では Eq. (3.3.2) で定義される有限インパルス応答型 RNN が用いられた。そのため、ここでは Type-1 と Type-2 の比較結果に焦点を当てる。

Table 3.6 から、検証データ集合に含まれるいずれの軌道に対しても、Type-1 の  $\mathcal{L}^{(\text{traj})}$  の値は Type-2 のものより小さいことがわかる。つまり、有限インパルス応答型 RNN の検証データ集合に対する予測精度は無限インパルス応答型 RNN に比べて高い精度であった。

### (2) 訓練手法の比較

3.2.1 節で述べた回帰分析ベースおよび 3.2.2 節で述べた軌道推定ベースの操縦モデルの訓練手法の比較結果について述べる。Type-1 と Type-3 では、同一の操縦モデルおよび訓練データ集合が用いられたが、訓練方法がそれぞれ異なる。Type-1 では軌道推定ベース、Type-3 では回帰分析ベースの訓練手法が用いられた。そのため、ここでは Type-1 と Type-3 の比較結果に焦点を当てる。

Table 3.6 から、検証データ集合に含まれるいずれの軌道に対しても、Type-3 は Type-1 に比べて極めて  $\mathcal{L}^{(\text{traj})}$  の値が大きく、特に、ランダム操船の結果に対して大きくなっていることがわかる。つまり、軌道予測ベースの手法で訓練された操縦モデルは回帰分析ベースの手法で訓練されたものと比べて、検証データ集合に対して高い予測精度を持っていた。ここで、訓練された操縦モデルの予測結果をより詳細に示すため、予測された操縦運動軌道を Fig. 3.4 に示す。Fig. 3.4 から、Type-3 では時刻が 100 s から 200 s の間に誤差が累積し、船体の姿勢変数ベクトル  $\eta$  に関する誤差が指数的に増加していることがわかる。その結果、Type-3 にお

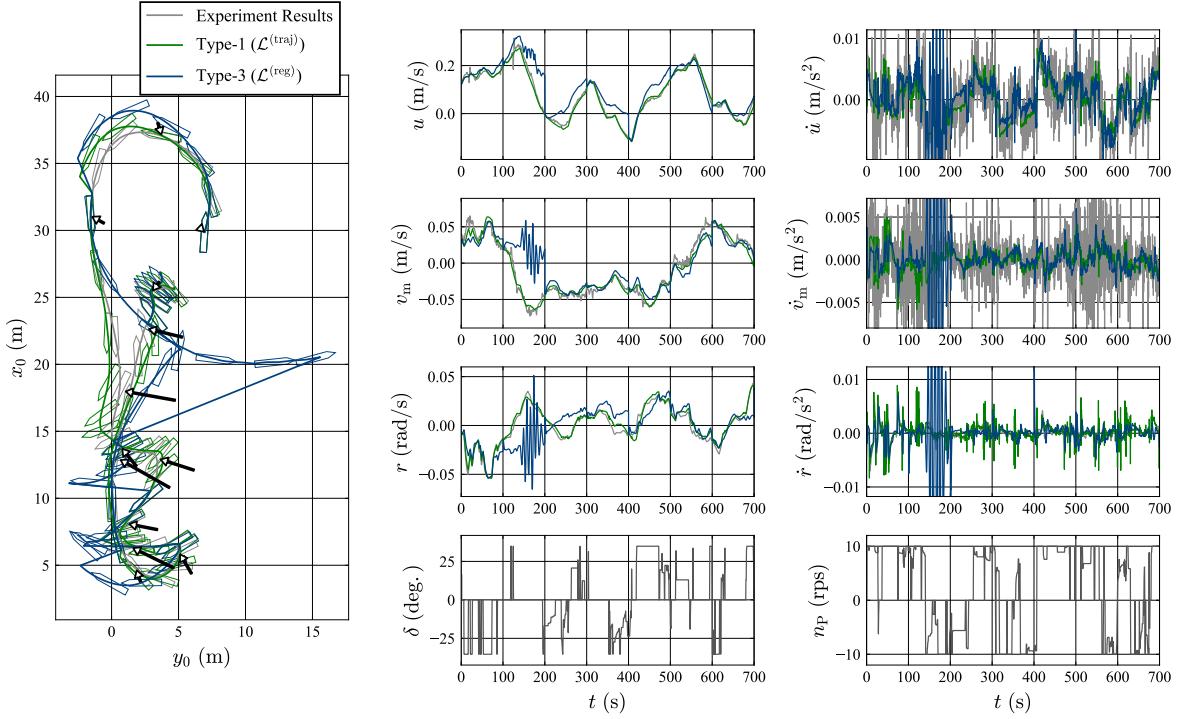


Fig. 3.4: Predicted trajectories of random maneuver using trained maneuvering model.

いて  $\mathcal{L}^{(traj)}$  の値が大きく評価されていることがわかる。

また, Fig. 3.4から, 船体の運動学的変数ベクトル  $\mathbf{x}$  に関しては高い精度で予測できているものの, 船体の加速度変数ベクトル  $\dot{\mathbf{v}}$  については高精度な予測が達成されていないことが確認できる. 2.3.1 節で述べたように, 船体の縦速度  $u$  および横速度  $v_m$  は, GNSS の位置座標データを数値的に時間微分することで求められている. さらに, 縦加速度  $\dot{u}$  および横加速度  $\dot{v}_m$  は, それぞれの速度データに対してさらに数値的な時間微分を行うことで観測されている. このため, 本実験で使用された加速度データにはノイズやスパイクが含まれる可能性が高く, Fig. 3.4からもその影響が読み取れる.

### (3) 訓練データ集合の比較

訓練データ集合の比較結果について述べる. Type-1, Type-4 および Type-5 では, 同一の操縦モデルおよび訓練手法が用いられたが, 訓練データ集合がそれぞれ異なる. Type-1 の訓練データ集合である Train-TZRB+ は Type-4 および Type-5 の訓練データ集合にくらべ約 5 倍以上のデータ量を持っている. また, Type-4 および Type-5 の訓練データ集合は同程度のデータ量を持つが, Type-5 にはランダム操船試験データが含まれており, Type-4 には含まれない.

Table 3.6から, 検証データ集合に含まれるいずれの軌道に対しても, Type-1 は Type-4 および Type-5 に比べて  $\mathcal{L}^{(traj)}$  の値が小さいことがわかる. このことから, データ量の多さに応じて操縦モデルの予測精度が向上することがわかる. ANN モデルは高い表現能力を持っているため, 訓練データ集合が多いほど訓練されていない操縦運動が減少し, 結果として検証データ集合に対する予測精度が向上していると考えられる. しかし, 必ずしも大量のデータ量を持つ訓練データ集合が与えられるわけではないことに注意が必要である.

Type-4 と Type-5 のみに焦点を当てると、Type-4 は Type-5 に比べてジグザク操船および旋回操船に対する  $\mathcal{L}^{(traj)}$  の値が小さくなっているが、ランダム操船に対しては逆に大きくなっていることがわかる。この結果は、Type-4 の訓練データ集合である Train-TZB には、Train-TZRB に比べ、より多くのジグザク操船および旋回操船のデータが含まれており、Type-5 の訓練データ集合である Train-TZRB には、Train-TZB には含まれない、ランダム操船のデータが含まれているためであると考えられる。しかし、Train-TZB と Train-TZRB に含まれる着桟操船のデータ量は同じであるにもかかわらず、Type-5 は Type-4 に比べて着桟操船に対する  $\mathcal{L}^{(traj)}$  の値が小さくなっていることがわかる。

#### (4) MMG モデルとの比較

最後に、MMG モデルとの比較結果について述べる。ここでは、拘束模型試験および経験式によって係数が決定された MMG モデルの結果と、Type-1 および Type-5 の結果に焦点を当てる。

Table 3.6から、ランダム操船、旋回操船、ジグザグ操船に対しては、Type-1 と Type-5 のいずれも MMG モデルに比べて  $\mathcal{L}^{(traj)}$  の値が小さいことがわかる。このことから、訓練データ集合に類似するデータが含まれていれば、MMG モデルより高い精度で操縦運動を予測可能であることがわかる。一方で、着桟操船では、必ずしも Type-1 と Type-5 が MMG モデルに比べ  $\mathcal{L}^{(traj)}$  の値が小さいわけではなく、 $B_1$  および  $B_3$  では MMG モデルがより小さい値を獲得していることがわかる。このことから、ランダム操船試験により着桟操船に対する予測精度を向上させる可能性があるが、本実験で用意された訓練データ集合では着桟操船の操縦運動を十分に網羅することができなかった可能性がある。また、MMG モデルおよび訓練された操縦モデルの予測結果をより詳細に示すため、予測された着桟操船の運動軌道を Fig. 3.5 に示す。Fig. 3.5 から、Type-1 の予測精度は訓練時のランダム性に依存しており、訓練された操縦モデルの全てが MMG モデルに劣っているわけではないことがわかる。

### 3.4 議論

3.3.3 節で得られた結果と結果から判明した点を以下にまとめる。

- 無限インパルス応答型 RNN は有限インパルス応答型 RNN に比べより長い期間の履歴影響を考慮した RNN モデルであるが、検証データ集合に対する予測精度が有限インパルス応答型 RNN に比べて向上はしなかった。よって、より長い期間の履歴影響を考慮することが、必ずしも操縦モデルの軌道推定精度を向上させるわけではないことがわかる。逆に言えば、必要以上に遠い過去の履歴影響を考慮することが、軌道推定精度を低下させる可能性があることを示している。例えば、ジグザグ操船は周期的な操縦運動であるため、遠い過去の履歴影響を考慮した RNN はジグザグ操船の周期性までも学習することが可能である。しかし、船舶の操縦モデルにはそのような周期性の学習は必要ではない。したがって、操縦モデルが考慮すべき履歴の長さは物理的に影響を受ける可能性のある有限値であることがより適切であると考えられる。
- 軌道推定ベースの提案訓練手法によって得られた操縦モデルは、回帰分析ベースの訓練手法によるものと比較して、検証データ集合に対する予測精度が大幅に向上した。本実験では、縦加速度  $\dot{u}$  および横加速度  $\dot{v}_m$  の観測値が数値時間微分によりノイズやスパイクの影響を受けやすく、加速度に関する計測値の使用が操縦モデルの予測精度を低下させる要因となっていたと考えられる。一方で、加速度計測値を用いて操縦モデルを推定可能な軌道推定ベースの訓練手法は、計測ノイズの影響を比較的軽減でき、

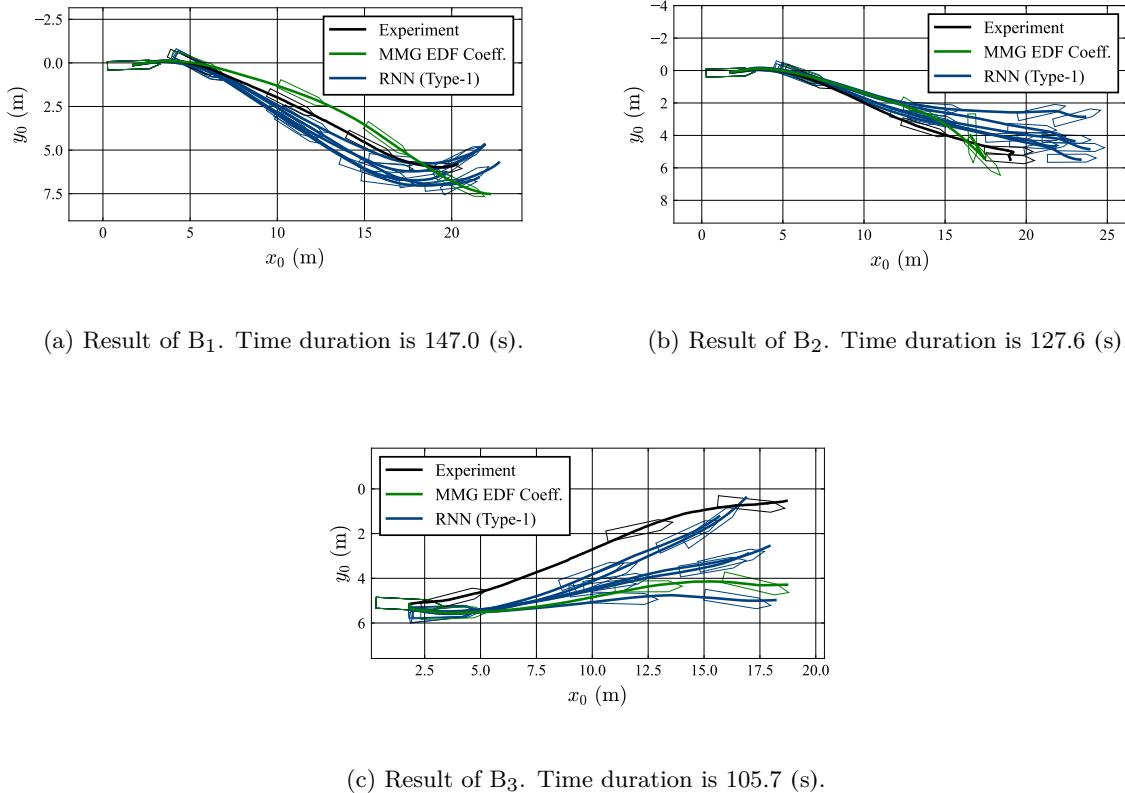


Fig. 3.5: Predicted trajectories of berthing maneuver using MMG model and trained maneuvering model.

その結果、操縦モデルの操縦運動予測精度が大幅に向上したと考えられる。

- より多くの訓練データを用いることで操縦モデルの予測精度が向上すること、また、ランダム操船試験データを用いることで、ジグザク操船および旋回操船に比べて、低速操縦運動を含む着桟操船軌道の予測精度を向上させることができた。一方で、ランダム操船試験はジグザク操船および旋回操船の検証データ集合に対する予測精度を向上させることはなかった。よって、検証データ集合に近いデータで訓練されていればRNNモデルの予測精度は向上し、低速操縦運動を効率的に計測するランダム操船試験が着桟操船のための操縦モデルの同定において有効なデータ収集方法であることがわかった。
- RNNで表現された操縦モデルは、拘束模型試験や経験式から得られたMMGモデルに比べ、検証データ集合のランダム操船、旋回操船、ジグザク操船に対する予測精度が高かったが、着桟操船では必ずしもRNNモデルの予測精度が高いとは限らなかった。よって、分布が近い訓練データが与えられれば、RNNモデルは拘束模型試験や経験式から得られたMMGモデルと同等かそれ以上に高い精度で、船舶の低速操縦運動を推定可能であることが分かった。

しかし、本実験では、高精度な操縦モデルの獲得のため、十分な量の操縦運動データを使用している。そのため、提案手法をより現実的で精度の高いものにするためには、適切な訓練データの十分な量の準備が不可欠である。この点の詳細な検討は今後の課題の一つである。

### 3.5 結言

本章では、ANNを用いた港内操船のための操縦モデルの推定手法を提案し、模型船の自由航走試験データを利用した検証実験結果を示した。本研究では、主に以下の5つの成果が得られた。

- 加速度の計測値を必要としない軌道推定ベースの訓練手法をRNNを用いた操縦モデルへ導入し、従来の回帰分析ベースの訓練手法によるものと比較して、検証データ集合に対する予測精度が大幅に向上去ることを示した。
- 低速操縦運動の予測精度を向上させるため、標準的な操縦運動を計測する旋回試験やジグザグ試験に加えて、ランダム操船試験を操縦モデルの推定のために使用した。その結果、低速操縦運動を含む着桟操船軌道の予測精度が向上することを示した。
- 有限インパルス応答型RNNに比べより長い期間の履歴影響を考慮した無限インパルス応答型RNNが、必ずしも操縦モデルの軌道推定精度を向上させるわけではなく、むしろ、軌道推定精度を低下させる可能性があることを示した。
- RNNを用いた操縦モデルは、分布が近い訓練データが与えられれば、拘束模型試験や経験式から得られたMMGモデルと同等かそれ以上に高い精度で、船舶の低速操縦運動を推定可能であることを示した。

しかし、この手法は高精度な操縦モデルの獲得のために十分な量の操縦運動データを必要とする。1.3節で述べたように必ずしも大量の操縦運動データを収集できるとは限らない。そのため、提案手法をより実用的にするためには、使用可能な操縦運動データが限られている場合の検討が必要である。

## 第 4 章

# 人工ニューラルネットワーク用いた操縦モ<sup>デル</sup>推定のためのデータ拡張手法

本章では、前節の ANN を用いた操縦モデルの推定手法に導入するデータ拡張手法について述べる。4.1 節では、本研究で導入するいくつかのデータ拡張手法について詳述し、4.2 節で模型船の自由航走試験データを用いた操縦モデルの推定実験の結果を示し、3.4 節で結果を元に議論を行う。最後に、3.5 節では本章の結言について述べる。

### 4.1 データ拡張手法

データ拡張とは、元のデータ集合を変換することで合成データを生成することである。本研究では、先行研究 [98] で紹介されているランダム変換に基づくデータ拡張のうち、空間および時間領域の変換に焦点を当てる。空間および時間領域の変換に基づくデータ拡張は、それぞれ、時系列の値か時間軸を変換することで行われ、ジッタリング (Jittering), 回転 (Rotation), スケール変換 (Scaling), 空間の変形 (Magnitude warping), スライシング (Slicing), 並び替え (Permutation), 時間の変形 (Time warping) などがある。

このうち、船舶の操縦モデル推定において適用可能な手法としてスライシングとジッタリングが考えられる。スライシングは元の時系列データから連続したスライスを抽出して合成データを生成する手法であるが、時系列データの時間変化量を変化させないので、オリジナルデータの意味を歪ませることはない。また、ジッタリングは時系列データにノイズを加えて合成データを生成する手法である。ジッタリングは、観測精度程度の大きさのノイズを用いることで、センサーデータの異なる実現に似たデータを生成することができる。ジッタリングもまた、適切なノイズを用いていれば、オリジナルデータを大きく歪ませることはない。そのため、本研究ではデータ拡張手法としてスライシングとジッタリングを用いた。

本研究では、Fig. 4.1に示すような、データ拡張を行わない部分列で構成される操縦運動データ集合、スライシングを用いて拡張された操縦運動データ集合、ジッタリングを用いて拡張された操縦運動データ集合を考える。これらの操縦運動データ集合に関する定義をそれぞれ4.1.1 節、4.1.2 節、4.1.3 節で詳述する。

#### 4.1.1 部分列

まず、データ拡張を行わない基準となるデータ集合を定義する。3.2.2 節では、一定のタイムステップ数  $K$  を持つ時系列データが与えられると仮定される。しかし、自由航走試験から得られる時系列データの計測時間

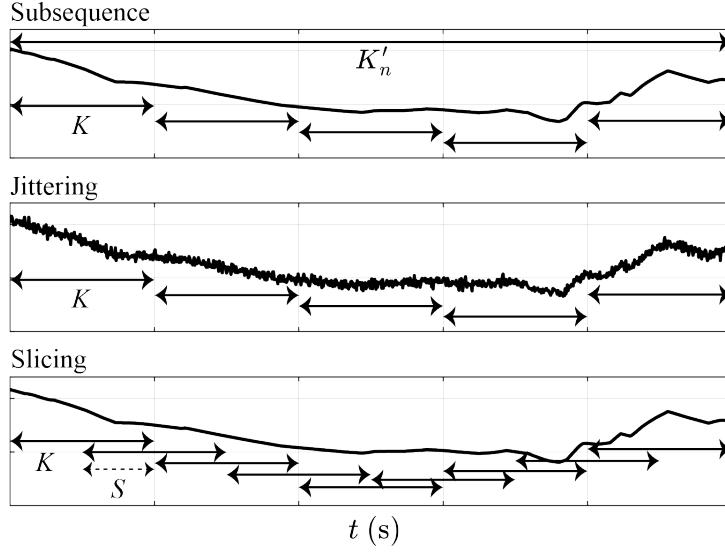


Fig. 4.1: Illustration of data augmentation methods.

は必ずしも一定とは限らない。そのため、実用的には、計測された時系列データが一定のタイムステップ数  $K$  で分割され、分割された部分列によってデータ集合がしばしば構成される。

ここで、 $N$  個の時系列データが与えられ、 $n$  番目の時系列データはある時刻  $t_n$  を初期時刻とし、 $\Delta t$  の時間間隔を持つ  $K'_n (> K)$  個の時刻点  $\{t_n + k\Delta t\}_{k=0}^{K'_n-1}$  で計測されたとする。この時、 $n$  番目の時系列データは  $\lfloor K'_n/K \rfloor$  個に分割することが可能で、分割された  $i$  番目の時系列データの初期時刻を  $t_n + iK\Delta t$  とすると、 $\mathbf{x}$  と  $\mathbf{u}$ 、 $\mathbf{w}_T$  に関する  $n$  番目の時系列データから分割された  $i$  番目の計測変数列はそれぞれ

$$\left\{ \begin{array}{l} \mathbf{X}_{K,n,i}^{(\text{sub})} = \left( \mathbf{x}^\top(t_n + iK\Delta t), \mathbf{x}^\top(t_n + (iK+1)\Delta t), \dots, \mathbf{x}^\top(t_n + (iK+K-1)\Delta t) \right)^\top \in \mathbb{R}^{6K} \\ \mathbf{U}_{K,n,i}^{(\text{sub})} = \left( \mathbf{u}^\top(t_n + iK\Delta t), \mathbf{u}^\top(t_n + (iK+1)\Delta t), \dots, \mathbf{u}^\top(t_n + (iK+K-1)\Delta t) \right)^\top \in \mathbb{R}^{N_u K} \\ \mathbf{W}_{T,K,n,i}^{(\text{sub})} = \left( \mathbf{w}_T^\top(t_n + iK\Delta t), \mathbf{w}_T^\top(t_n + (iK+1)\Delta t), \dots, \mathbf{w}_T^\top(t_n + (iK+K-1)\Delta t) \right)^\top \in \mathbb{R}^{2K} \end{array} \right. \quad (4.1.1)$$

と表される。このとき、Eq. (3.2.4) のデータ集合は

$$\mathcal{D}^{(\text{traj})} = \left\{ \left\{ \left( \mathbf{X}_{K,n,i}^{(\text{sub})}, \mathbf{U}_{K,n,i}^{(\text{sub})}, \mathbf{W}_{T,K,n,i}^{(\text{sub})} \right) \right\}_{i=0,1,\dots,\lfloor \frac{K'_n}{K} \rfloor-1} \right\}_{n=1,\dots,N} \quad (4.1.2)$$

と再定義することが可能である。

#### 4.1.2 スライシング

次に、スライシングによるデータ集合の拡張方法を示す。スライシングは元の時系列データから一定長さの時系列データを抜き出すことでデータを生成する。そして、抜き出す時系列データはランダムに選択されるか [99]、開始時間ステップをスライドさせて選択される [123]。スライシングを用いると、異なる時系列データ間でデータの重複して使用するため、より多くの時系列データを生成することが可能となる。特に、本論文で扱うモデル同定手法では、モデルの推定精度は運動学的変数の初期値に含まれる観測誤差に大きく影響を受ける。スライシングは特定の初期値に含まれる観測誤差の影響を相対的に小さくできる可能性を持つ。

ここで、4.1.1節と同様に、 $N$ 個の時系列データが与えられ、 $n$ 番目の時系列データはある時刻 $t_n$ を初期時刻とし、 $\Delta t$ の時間間隔を持つ $K'_n (> K)$ 個の時刻点 $\{t_n + k\Delta t\}_{k=0}^{K'_n-1}$ で計測されたとする。本研究では、初期時刻を一定のタイムステップ間隔 $S$ でスライドさせながら、一定のタイムステップ数 $K$ を持つ時系列データを抽出する。この時、 $n$ 番目の時系列データから $\lfloor(K'_n - K)/S\rfloor$ 個だけ抽出することが可能で、抽出された $i$ 番目の時系列データの初期時刻を $t_n + iS\Delta t$ とすると、 $\mathbf{x}$ と $\mathbf{u}$ 、 $\mathbf{w}_T$ に関する $n$ 番目の時系列データから抽出された $i$ 番目の計測変数列はそれぞれ

$$\begin{cases} \mathbf{X}_{K,n,i}^{(\text{sli})} = \left( \mathbf{x}^\top(t_n + iS\Delta t), \mathbf{x}^\top(t_n + (iS+1)\Delta t), \dots, \mathbf{x}^\top(t_n + (iS+K-1)\Delta t) \right)^\top \in \mathbb{R}^{6K} \\ \mathbf{U}_{K,n,i}^{(\text{sli})} = \left( \mathbf{u}^\top(t_n + iS\Delta t), \mathbf{u}^\top(t_n + (iS+1)\Delta t), \dots, \mathbf{u}^\top(t_n + (iS+K-1)\Delta t) \right)^\top \in \mathbb{R}^{N_u K} \\ \mathbf{W}_{T,K,n,i}^{(\text{sli})} = \left( \mathbf{w}_T^\top(t_n + iS\Delta t), \mathbf{w}_T^\top(t_n + (iS+1)\Delta t), \dots, \mathbf{w}_T^\top(t_n + (iS+K-1)\Delta t) \right)^\top \in \mathbb{R}^{2K} \end{cases} \quad (4.1.3)$$

と表現できる。このとき、Eq. (3.2.4) のデータ集合は

$$\mathcal{D}^{(\text{traj})} = \left\{ \left\{ \mathbf{X}_{K,n,i}^{(\text{sli})}, \mathbf{U}_{K,n,i}^{(\text{sli})}, \mathbf{W}_{T,K,n,i}^{(\text{sli})} \right\}_{i=0,1,\dots,\lfloor \frac{K'_n-K}{S} \rfloor} \right\}_{n=1,2,\dots,N} \quad (4.1.4)$$

と再定義することが可能である。

### 4.1.3 ジッタリング

ここでは、ジッタリングによるデータ集合の拡張方法を示す。ジッタリングは時系列データに対してノイズを加えることで合成データを生成する。ニューラルネットワークでは入力にノイズを加えることにより汎化性能を向上させることで知られている[124]。また、ジッタリングは時系列データにノイズが含まれていることを前提としており、この手法はセンサーデータに適用されることが多い[100, 125]。

本研究では、時間及び空間に対して独立なゼロ平均正規分布に従うノイズを船体の運動学的変数ベクトル $\mathbf{x}$ に加える。船体の運動学的変数ベクトル $\mathbf{x}$ の成分毎に加える正規ノイズの標準偏差をそれぞれ $\sigma_{x_0}^{(\text{jit})}, \sigma_{y_0}^{(\text{jit})}, \sigma_\psi^{(\text{jit})}, \sigma_u^{(\text{jit})}, \sigma_{v_m}^{(\text{jit})}, \sigma_r^{(\text{jit})}$ と表記すると、 $j$ 番目の正規ノイズを

$$\mathbf{E}_j \sim \mathcal{N} \left( \mathbf{0}, \text{diag} \left( \left( \sigma_{x_0}^{(\text{jit})} \right)^2, \left( \sigma_{y_0}^{(\text{jit})} \right)^2, \left( \sigma_\psi^{(\text{jit})} \right)^2, \left( \sigma_u^{(\text{jit})} \right)^2, \left( \sigma_{v_m}^{(\text{jit})} \right)^2, \left( \sigma_r^{(\text{jit})} \right)^2 \right) \right) \quad (4.1.5)$$

と定義する。この時、ジッタリングにより $J$ 通りのノイズをEq. (4.1.2) のデータ集合へ加える場合、拡張されたデータ集合は

$$\mathcal{D}^{(\text{traj})} = \left\{ \left\{ \mathbf{X}_{K,n,i}^{(\text{sub})} + \mathbf{E}_j, \mathbf{U}_{K,n,i}^{(\text{sub})}, \mathbf{W}_{T,K,n,i}^{(\text{sub})} \right\}_{i=0,1,\dots,\lfloor K'_n/K \rfloor-1} \right\}_{n=1,\dots,N, j=1,2,\dots,J} \quad (4.1.6)$$

と定義される。また、同様にノイズをEq. (4.1.4) のデータ集合へ加える場合、拡張されたデータ集合は

$$\mathcal{D}^{(\text{traj})} = \left\{ \left\{ \mathbf{X}_{K,n,i}^{(\text{sli})} + \mathbf{E}_j, \mathbf{U}_{K,n,i}^{(\text{sli})}, \mathbf{W}_{T,K,n,i}^{(\text{sli})} \right\}_{i=0,1,\dots,\lfloor \frac{K'_n-K}{S} \rfloor} \right\}_{n=1,2,\dots,N, j=1,2,\dots,J} \quad (4.1.7)$$

と定義される。この手法では、発生するノイズの数に比例してデータを増やすことができ、所望の大きさのノイズを加えることが可能である。しかし、不必要に大きなノイズは推定精度を悪化させる。そこで、本研究では、測定器の観測精度に応じて、付加するノイズの標準偏差を決定することとした。

Table 4.1: Trajectory data collected by free-running model tests. The sampling frequency is 1 Hz.

Trajectory No.	Duration	Maneuver
No. 1	500.5 (s)	Random
No. 2	1801.8 (s)	Random
No. 3	500.5 (s)	Random
No. 4	1801.8 (s)	Random
No. 5	1201.2 (s)	Random
No. 6	1201.2 (s)	Random
No. 7	100.0 (s)	Berthing
No. 8	100.0 (s)	Berthing

## 4.2 模型試験データを用いた検証実験

本節では、4.1 節で述べたデータ拡張手法の有効性を示すため、軌道推定ベースの訓練手法にデータ拡張手法を組み込んだ場合の船舶操縦モデルの推定実験の結果を示す。ただし、操縦モデルは3.1 節で述べた FNN モデルによって表現される。ここで示す内容は、自著論文 2 [33] で示された内容と同様である。本実験では、2.4 節で述べた供試船 2 が対象船舶とされた。以降、4.2.1 節で使用された操縦運動データ集合について詳述し、4.2.2 節で訓練結果について、4.2.3 節で操縦運動の推定結果について示す。

### 4.2.1 操縦運動データ集合

本実験で用いられる操縦運動データ集合は、2.5 節で述べた大阪大学の実験池である犬飼池にて実施された供試船 2 の自由航走試験により収集された。自由航走試験の詳細を以降に述べる。

#### (1) データ集合

本実験では、ランダム操船試験と仮想岸壁を考慮した手動着桟操船試験で計測された操縦運動データ集合を使用する。3.3.1 節で述べたように、ランダム操船試験では、港内操船のような低速状態での大きな斜航角を持つ運動を含む、広範囲に分布するデータを測定することを目的とし、操船者が制御入力を手動で選択する。ただし、プロペラ回転数は  $n_P$  (rps)  $\in [0.0, 12.5]$  の範囲で選択され、バウスラスタ回転数は  $n_{BT} = 0.0$  (rps) で固定された。

計測された操縦運動データ集合を Table 4.1 に示す。ただし、自由航走試験では 10 Hz の周波数で計測されたが、操縦運動データ集合は 1 Hz にダウンサンプリングされたデータで構成されていることに注意する。計測された軌道のうち、軌道 1 から 4 は訓練データ集合のため、軌道 5 は評価データ集合のため、軌道 6 から 8 は検証データ集合のために使用された。本実験では、データ拡張手法の適用の有無による違いを確認するため、Table 4.2 に示すように 8 種類の訓練データ集合を用意した。 $\mathcal{D}^{(sub)}$  はデータ拡張手法を用いない基準となる訓練データ集合  $\mathcal{D}^{(sli2)}$  及び  $\mathcal{D}^{(jit2)}$  はそれぞれのデータ拡張手法により 2 倍のデータ量、 $\mathcal{D}^{(sli10)}$  及び  $\mathcal{D}^{(jit10)}$  は 10 倍のデータ量に拡張した訓練データ集合である。また、 $\mathcal{D}^{(sli2 \times jit2)}$  はスライシングおよびジッタリングを併用して 4 倍のデータ量、 $\mathcal{D}^{(sli10 \times jit10)}$  は 100 倍のデータ量に拡張されている。また、 $\mathcal{D}^{(d-sub)}$

Table 4.2: Training, validation and test datasets.

Symbol	Trajectory No.	Augmentation method	Dataset definition
$\mathcal{D}^{(\text{sub})}$	No. 1 and 2	Subsequence	Eq. (4.1.2) where $K = 100$
$\mathcal{D}^{(\text{sli2})}$	No. 1 and 2	Slicing	Eq. (4.1.4) where $K = 100, S = 50$
$\mathcal{D}^{(\text{sli10})}$	No. 1 and 2	Slicing	Eq. (4.1.4) where $K = 100, S = 10$
$\mathcal{D}^{(\text{jit2})}$	No. 1 and 2	Jittering	Eq. (4.1.6) where $K = 100, J = 2$
$\mathcal{D}^{(\text{jit10})}$	No. 1 and 2	Jittering	Eq. (4.1.6) where $K = 100, J = 10$
$\mathcal{D}^{(\text{sli2} \times \text{jit2})}$	No. 1 and 2	Slicing and Jittering	Eq. (4.1.7) where $K = 100, S = 50, J = 2$
$\mathcal{D}^{(\text{sli10} \times \text{jit10})}$	No. 1 and 2	Slicing and Jittering	Eq. (4.1.7) where $K = 100, S = 10, J = 10$
$\mathcal{D}^{(\text{d-sub})}$	No. 1, 2, 3 and 4	Subsequence	Eq. (4.1.2) where $K = 100$
$\mathcal{D}^{(\text{valid})}$	No. 5	Subsequence	Eq. (4.1.2) where $K = 100$
$\mathcal{D}^{(\text{test-R})}$	No. 6	Subsequence	Eq. (4.1.2) where $K = 100$
$\mathcal{D}^{(\text{test-B})}$	No. 7 and 8	Subsequence	Eq. (4.1.2) where $K = 100$

はデータ拡張手法を用いず 2 倍のデータ量を持つ訓練データ集合である。

用意された訓練データ集合、評価データ集合、検証データ集合の分布を確認するため、船体の速度変数ベクトル  $v$  のヒストグラムを Fig. 4.2 に、アクチュエータ状態変数ベクトル  $u$  の散布図を Fig. 4.3 に、相対風速風向ベクトル  $w_A$  の二次元ヒストグラムを Fig. 4.4 に示す。Fig. 4.2 から、 $\mathcal{D}^{(\text{sub})} + \mathcal{D}^{(\text{valid})}$  および  $\mathcal{D}^{(\text{d-sub})} + \mathcal{D}^{(\text{valid})}$  のいずれも  $\mathcal{D}^{(\text{test-B})}$  をカバーできているが、 $\mathcal{D}^{(\text{test-R})}$  を完全にカバーできていないことがわかる。例えば、 $\mathcal{D}^{(\text{sub})} + \mathcal{D}^{(\text{valid})}$  は  $\mathcal{D}^{(\text{test-R})}$  に存在する 1.5 (m/s) 以上の横速度のデータをカバーできていない。また、 $\mathcal{D}^{(\text{test-R})}$  では  $U_A = 5.0$  (m/s),  $\gamma_A = 300$  (deg.) の風が発生しているが、 $\mathcal{D}^{(\text{sub})}$  にはそのようなデータはまったく含まれない。

本実験は模型船が用いられるため、検証データ集合を完全にカバーするデータを収集することは可能である。しかし、これは実スケールの船舶では容易ではない。特に、実船試験では風速を制御することが不可能であり、発生頻度の少ない強風データを収集することは難しい。よって、実用的には、状態変数の発生しうる全ての状態を訓練データ集合に含めることができないため、訓練データ集合に含まれない状態変数に遭遇する可能性は高い。そのため、本研究では訓練データ集合に含まれない外挿データに対する推定精度について検証を行っている。

## (2) 港内操船との比較

ランダム操船試験で計測されたデータの分布が、用意した着桟操船の操縦データのみならず、実際の港湾の操縦運動の分布の多くの範囲をカバーしていることを示すため、ランダム操船試験で計測されたデータと入出港時の実船計測データとの比較結果を示す。宮内らは、本研究で用いた対象船舶の港湾航行と操船の統計解析に関する研究を行っている [126]。彼らは、フルスケール船の入出港時の操船運動データを記録し、このデータを基に入港時と出港時の操船運動の統計的特徴を示している。よって、宮内らの解析結果と計測データの統計的特徴を比較する。ただし、ランダム操縦試験データは実船データとは縮尺が異なるため、相似則または無次元値を用いて比較する。

まず、着桟点付近における対地船速と斜航角について焦点を当てる。文献 [126] では、繫留位置からのユ

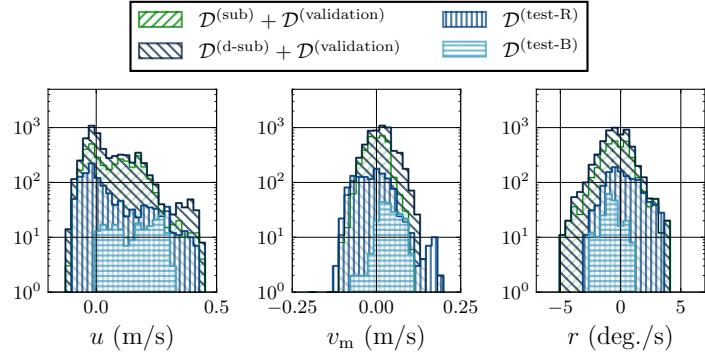


Fig. 4.2: Histograms of ship state variables. Note that the vertical axes, which show the frequency, are scaled logarithmically.

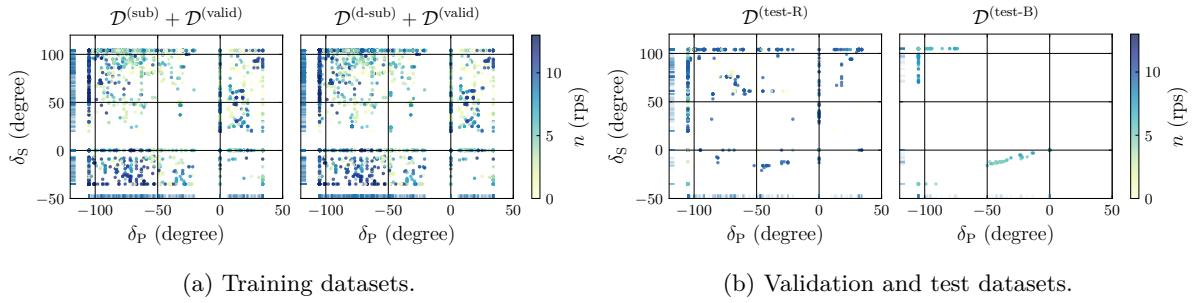


Fig. 4.3: Scatter plots of  $\delta_P$ ,  $\delta_S$ . The color bar shows the propeller revolution number  $n_P$ .

一クリッド距離が  $2.0L_{pp}$  より小さい場合、大斜航角が取られることが多いと示されている。そして、その範囲における対地速度  $U = \sqrt{u^2 + v_m^2}$  のヒストグラムが文献 [126] の Figure 5 に示されており、同様の図を Fig. 4.5aに示す。ただし、 $\beta$  は斜航角を示している。この結果から、離着桟操船では、 $U < 1.5$  (knot) の場合に大斜航角が取られることがわかる。

用意したデータ集合を用いて同様のヒストグラムを作成し、Fig. 4.5に示した。ただし、相似則により、速度がフルスケールに変換された。Fig. 4.5から、ランダム操縦試験では低速の操縦データが測定され、 $U < 1.5$  (knot) の場合、斜航角の大きい操縦運動が測定されていることがわかる。

また、 $\beta$  と  $r$  の関係について焦点を当てる。文献 [126] では、 $r$  は小瀬の方法 [4] により無次元化され、その無次元値は  $r^* = r\sqrt{L_{pp}/g}$  と定義されている。 $r^*$  と  $\beta$  の散布図を [126] の Figure 6 に、同じ図を Fig. 4.6aに示す。Fig. 4.6aから、船速が速い場合、 $r$  と  $\beta$  の間には、斜航角の小さい範囲で強い正の相関が見られる。この特徴はジグザグや旋回のデータにも見られる。一方、1 ノット以下の低速域では、幅広い斜航角が測定されている。特に接岸データでは、無次元ヨー速度  $r^*$  は-0.02～0.02 の範囲に分布し、 $r$  と  $\beta$  の間には緩やかな傾きを持つ正の相関が見られる。

さらに、Figs. 4.6bと4.6cに  $r^*$  と  $\beta$  に関する操縦運動データ集合の散布図を示す。Figs. 4.6bと4.6cに示すように、 $\mathcal{D}^{(test-R)}$  も、船速が速い場合には  $r$  と  $\beta$  の間に強い正の相関があった。低速域では、幅広い斜航角の操縦運動データ集合が測定されている。特に、データ量の多い  $\mathcal{D}^{(d-sub)}$  の  $r^*$  は、どの斜航角でも広く分布している。

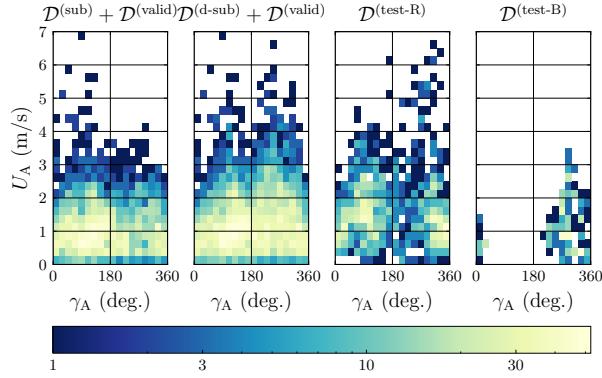


Fig. 4.4: 2D histograms of apparent wind speed and direction. The bin width of apparent wind speed is 0.25 (m/s), and that of apparent wind direction is 20 (deg.). The color bar shows the frequency with a logarithmic scale.

Table 4.3: The used network layers for maneuvering model.

	Dimension of $\mathbf{W}_l$	Dimension of $\mathbf{b}_l$	Activation function $g$
$\mathbf{h}_1^{(fcl)}$	$\mathbb{R}^{256 \times 7}$	$\mathbb{R}^{256}$	hyperbolic tangent function
$\mathbf{h}_2^{(fcl)}$	$\mathbb{R}^{256 \times 256}$	$\mathbb{R}^{256}$	hyperbolic tangent function
$\mathbf{h}_3^{(fcl)}$	$\mathbb{R}^{256 \times 256}$	$\mathbb{R}^{256}$	hyperbolic tangent function
$\mathbf{h}_4^{(fcl)}$	$\mathbb{R}^{256 \times 256}$	$\mathbb{R}^{256}$	hyperbolic tangent function
$\mathbf{h}_5^{(fcl)}$	$\mathbb{R}^{3 \times 256}$	$\mathbb{R}^3$	None

従って、計測されたデータには、 $r$  と  $\beta$  の間に強い相関を持っている船速が高い運動データのみならず、低速状態における広い範囲の斜航角を持つ運動も多く計測されており、計測されたデータの分布は港湾の操縦運動のフルスケールデータの分布の大部分をカバーしていた。

#### 4.2.2 実験内容

実施した訓練内容の詳細な設定について述べる。本実験では、全結合層のみで構成される多層パーセプトロンを用いて操縦モデルの推定が行われた。具体的に、使用された FNN モデルは

$$\mathbf{y}^{(\text{FNN})} = \mathbf{h}_5^{(\text{fcl})} \circ \mathbf{h}_4^{(\text{fcl})} \circ \mathbf{h}_3^{(\text{fcl})} \circ \mathbf{h}_2^{(\text{fcl})} \circ \mathbf{h}_1^{(\text{fcl})} (\mathbf{x}^{(\text{FNN})}) \quad (4.2.1)$$

と定義される。ここで、 $\mathbf{h}_1^{(\text{fcl})}, \mathbf{h}_2^{(\text{fcl})}, \mathbf{h}_3^{(\text{fcl})}, \mathbf{h}_4^{(\text{fcl})}, \mathbf{h}_5^{(\text{fcl})}$  は Table 4.3 で定義される全結合層である。

また、3.2.2 節で述べた軌道推定ベースの訓練手法が用いられたが、本実験で用いられた損失関数は Eq. (3.2.10) と少し異なる。本実験では、正則化項が加えられ、損失関数は

$$\begin{aligned} \mathcal{L}^{(\text{traj})} (\boldsymbol{\theta}; \mathcal{D}^{(\text{traj})}) &= \frac{1}{N} \sum_{n=1}^N \int_{t=t_n}^{t_n+(K-1)\Delta t} \|\mathbf{x}(t) - \mathbf{x}_{\boldsymbol{\theta}}(t)\|_{\mathbf{W}_x}^2 dt + \lambda \|\boldsymbol{\theta}\|^2 \\ &\approx \frac{1}{N} \sum_{n=1}^N \sum_{k=0}^{K-2} \frac{\|\mathbf{x}(t_n + (k+1)\Delta t) - \mathbf{x}_{\boldsymbol{\theta}}(t_n + (k+1)\Delta t)\|_{\mathbf{W}_x}^2 + \|\mathbf{x}(t_n + k\Delta t) - \mathbf{x}_{\boldsymbol{\theta}}(t_n + k\Delta t)\|_{\mathbf{W}_x}^2}{2} \Delta t + \lambda \|\boldsymbol{\theta}\|^2 \end{aligned} \quad (4.2.2)$$

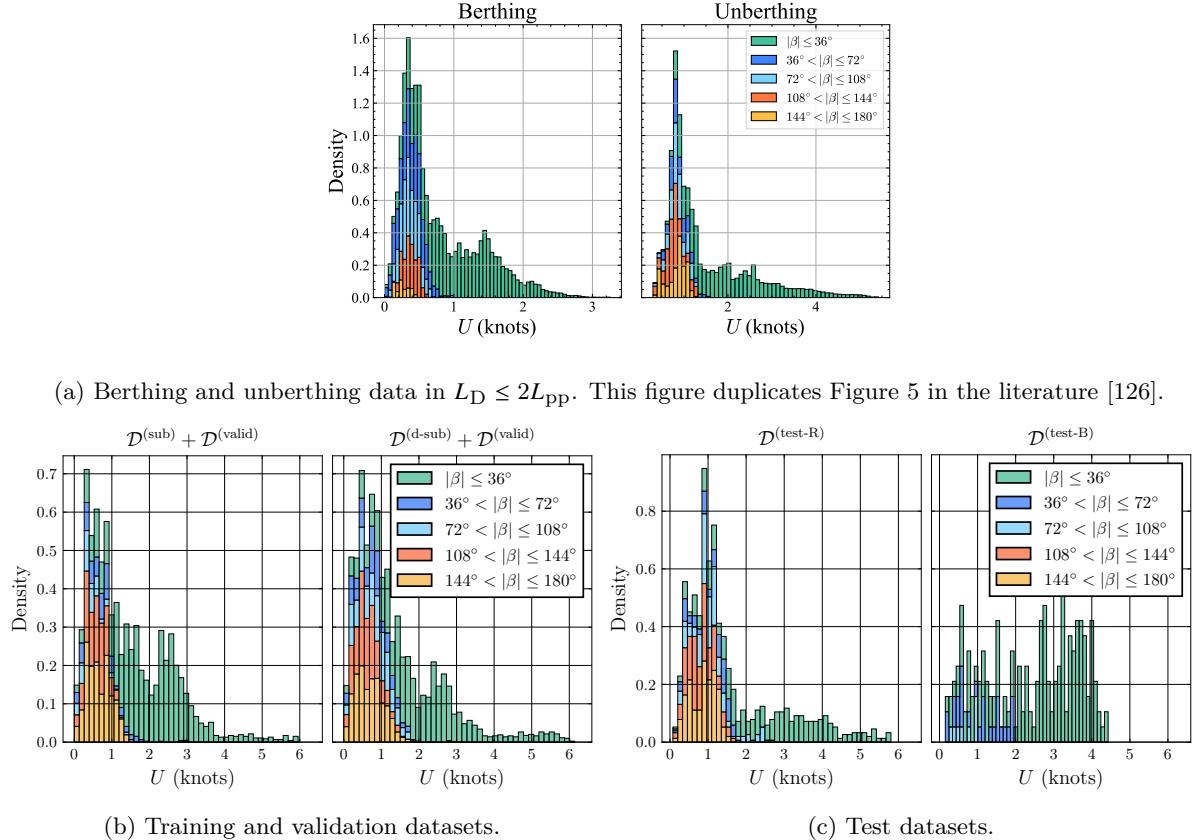


Fig. 4.5: Histograms of  $|\beta|$  for  $U$ . Histograms are normalized so that the sum of their areas equals 1 in each figure.

と定義された。ここで、 $\lambda$ は正則化パラメータである。損失関数における重み行列  $\mathbf{W}_x$  は速度成分に関する観測誤差の分散に応じた重み付け方法によって決定される。つまり、

$$\mathbf{W}_x = \text{diag} \left( \left( \sigma_{x_0}^{(\text{obs})} \right)^{-2}, \left( \sigma_{y_0}^{(\text{obs})} \right)^{-2}, \left( \sigma_{\psi}^{(\text{obs})} \right)^{-2}, \left( \sigma_u^{(\text{obs})} \right)^{-2}, \left( \sigma_{v_m}^{(\text{obs})} \right)^{-2}, \left( \sigma_r^{(\text{obs})} \right)^{-2} \right) \quad (4.2.3)$$

とされた。ここで、 $\sigma_{x_0}^{(\text{obs})}$ ,  $\sigma_{y_0}^{(\text{obs})}$ ,  $\sigma_{\psi}^{(\text{obs})}$ ,  $\sigma_u^{(\text{obs})}$ ,  $\sigma_{v_m}^{(\text{obs})}$ ,  $\sigma_r^{(\text{obs})}$  は成分毎の観測誤差に関する標準偏差であり、本実験では船体の姿勢変数ベクトル  $\boldsymbol{\eta}$  に関する観測誤差を無視するため  $\left( \sigma_{x_0}^{(\text{obs})} \right)^{-2} = \left( \sigma_{y_0}^{(\text{obs})} \right)^{-2} = \left( \sigma_{\psi}^{(\text{obs})} \right)^{-2} = 0.0$  とされた。

本実験では、Table 4.2に示す全ての訓練データ集合から得られた操縦モデルの推定精度に関する比較を行う。ただし、乱数を変更して最適化をそれぞれ 10 回実施され、最適なパラメータは各学習データセットと各乱数についてそれぞれ求められた。各エポックにおける検証データセットを使用した損失関数の指數移動平均値を Fig. 4.7に示す。ただし、損失関数の指數移動平均値は

$$\hat{\mathcal{L}}^{(\text{traj})} (\boldsymbol{\theta}_i; \mathcal{D}^{(\text{valid})}) = \begin{cases} \alpha \mathcal{L}^{(\text{traj})} (\boldsymbol{\theta}_i; \mathcal{D}^{(\text{valid})}) + (1 - \alpha) \hat{\mathcal{L}}^{(\text{traj})} (\boldsymbol{\theta}_{i-1}; \mathcal{D}^{(\text{valid})}) & (i \neq 0) \\ \mathcal{L}^{(\text{traj})} (\boldsymbol{\theta}_0; \mathcal{D}^{(\text{valid})}) & (i = 0) \end{cases}, \quad (4.2.4)$$

と定義される。ここで、 $\alpha = 0.1$  で、 $\boldsymbol{\theta}_i$  は  $i$  番目のエポックのパラメータベクトル、 $\hat{\mathcal{L}}^{(\text{traj})}$  は指數移動平均の

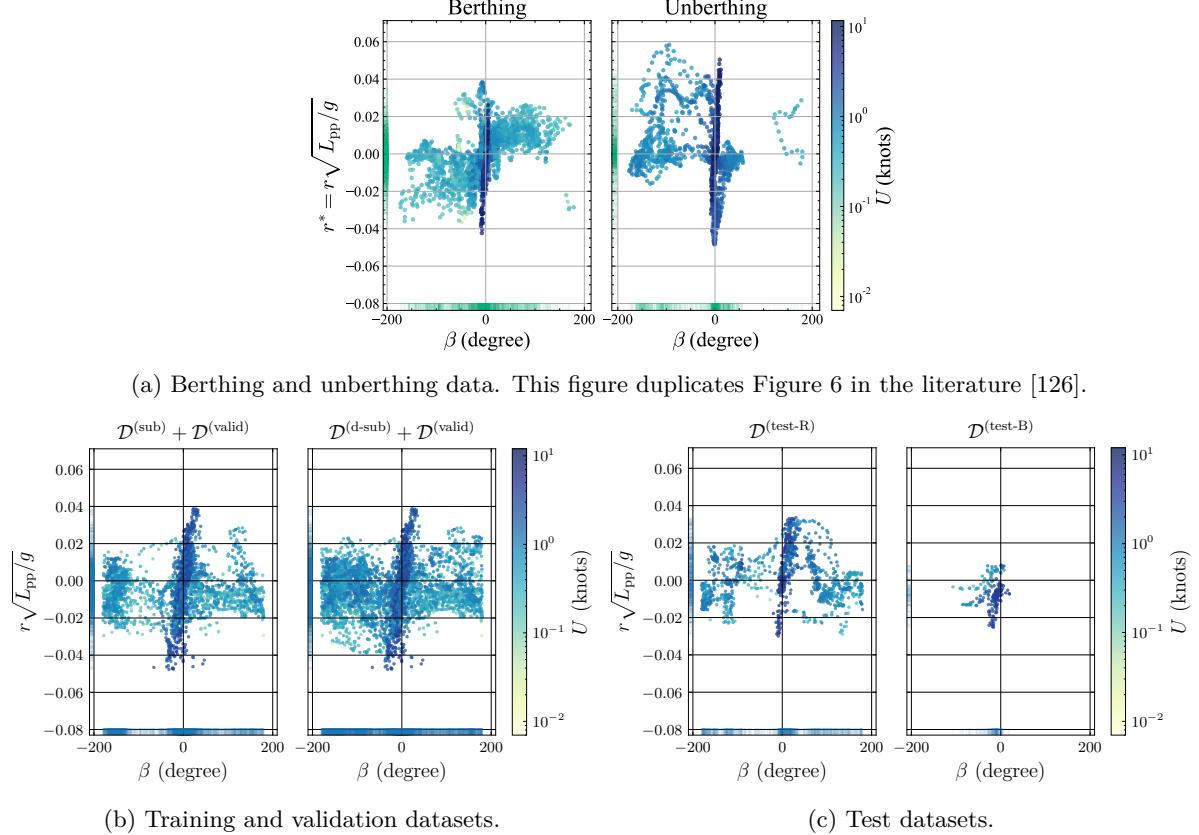
Fig. 4.6: Scatter plots of  $\beta$  and non-dimensionalized yaw velocity  $r^* = r\sqrt{L_{pp}/g}$ .

Table 4.4: Hyperparameters in training.

Item	Value
Learning rate	$1.0 \times 10^{-4}$
Regularization coefficient: $\lambda$	$1.0 \times 10^{-2}$
Duration of time step: $\Delta t$	1.0 (s)
Standard deviation of $u$ noise: $\sigma_u^{(\text{obs})}$ and $\sigma_u^{(\text{jit})}$	0.01 (m/s)
Standard deviation of $v_m$ noise: $\sigma_{v_m}^{(\text{obs})}$ and $\sigma_{v_m}^{(\text{jit})}$	0.01 (m/s)
Standard deviation of $r$ noise: $\sigma_r^{(\text{obs})}$ and $\sigma_r^{(\text{jit})}$	0.1 (deg./s)

損失関数値を意味する。また、訓練は 10,000 Epoch を超えた後、

$$\mathcal{L}^{(\text{traj})}(\theta_i; \mathcal{D}^{(\text{valid})}) > 0.1 \times \mathcal{L}^{(\text{traj})}(\theta_0; \mathcal{D}^{(\text{valid})}) + 0.9 \times \min_{0 \leq j \leq i} \mathcal{L}^{(\text{traj})}(\theta_j; \mathcal{D}^{(\text{valid})}) \quad (4.2.5)$$

を満たしたときに訓練が終了された。Fig. 4.7を見ると、ほとんどのケースで、Eq. (4.2.5) を満たすことで、訓練が終了していることがわかる。よって、このことから訓練データに対するオーバーフィッティングが発生していることがわかる。

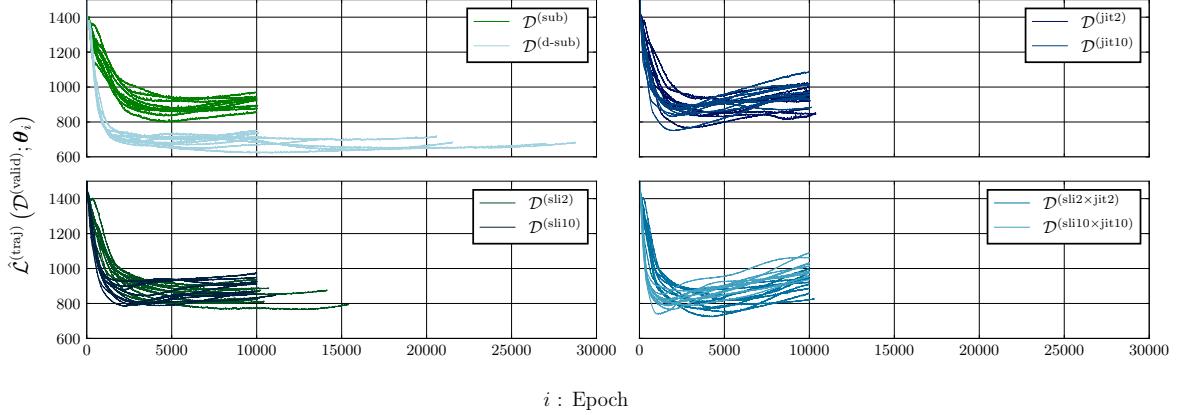


Fig. 4.7: Exponential moving average values of the evaluation function in the validation dataset. The legend implies the used training dataset. Ten training results with different random numbers for each dataset are presented.

### 4.2.3 実験結果

得られた最適パラメータを用いた操縦モデルの検証データ集合に対する予測精度を示す。検証データ集合における操縦モデルの予測誤差を表す損失関数を求めた。得られた損失関数の値と乱数に対するその平均値を Fig. 4.8 に示す。また、Fig. 4.9 に検証データ集合のアクチュエータ状態変数ベクトル  $\mathbf{u}$  及び相対風速風向ベクトル  $\mathbf{w}_A$  の時系列を示し、Figs. 4.10a–4.10d に予測された船体の速度変数ベクトル  $\mathbf{v}$  の時系列を示す。特に、Fig. 4.10a では  $\mathcal{D}^{(sub)}$  と  $\mathcal{D}^{(double)}$ 、Fig. 4.10b では  $\mathcal{D}^{(sub)}$  と  $\mathcal{D}^{(sli10)}$ 、Fig. 4.10c では  $\mathcal{D}^{(sub)}$  と  $\mathcal{D}^{(jit10)}$ 、Fig. 4.10d では  $\mathcal{D}^{(sub)}$  と  $\mathcal{D}^{(sli10 \times jit10)}$  の結果を示している。ただし、数値シミュレーションのタイムステップ数は訓練時と同様であり、 $K = 100$  毎に計測データに初期化されている。

スライシングで拡張された訓練データ集合について焦点を当てる。Fig. 4.8 を見ると、スライシングで拡張された  $\mathcal{D}^{(sli2)}$  と  $\mathcal{D}^{(sli10)}$  の損失関数値の平均値はデータ拡張されていない  $\mathcal{D}^{(sub)}$  のそれに比べて小さくなっていることがわかる。よって、スライシングが検証データ集合に対する予測精度を向上させていることがわかる。しかし、 $\mathcal{D}^{(sli10)}$  と  $\mathcal{D}^{(sli2)}$  の差が小さいことから、データ拡張量を大きくすればするほど予測精度が必ずしも向上するわけではないことがわかる。

ジッタリングで拡張された訓練データ集合について焦点を当てる。ジッタリングで拡張された  $\mathcal{D}^{(jit2)}$  は  $\mathcal{D}^{(sub)}$  の損失関数値の平均値に比べて少しだけ小さくなっている。よって、ジッタリングもまた検証データ集合に対する予測精度を向上させていることが分かる。

スライシングとジッタリングが併用された訓練データ集合について焦点を当てる。 $\mathcal{D}^{(sli2 \times jit2)}$  および  $\mathcal{D}^{(sli10 \times jit10)}$  もまたの損失関数値の平均値はデータ拡張されていない  $\mathcal{D}^{(sub)}$  のそれに比べて小さくなっていることがわかる。また、スライシングとジッタリングのいずれか一方を使用した場合に比べ、少しではあるが損失関数値の平均値が減少していることがわかる。よって、スライシングとジッタリングは併用しても検証データ集合に対する予測精度を向上させていることが分かる。

しかし、データ拡張を行なったどのデータ集合よりも  $\mathcal{D}^{(d-sub)}$  の損失関数値が小さい。つまり、 $\mathcal{D}^{(sli10)}$  や

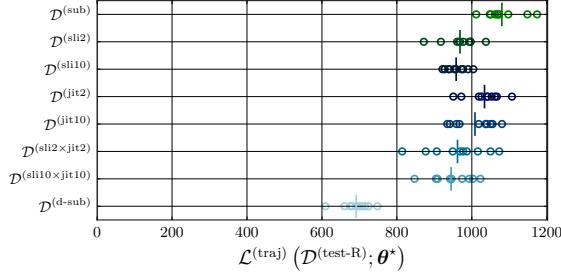


Fig. 4.8: Evaluation function values for random maneuver  $\mathcal{D}^{(\text{test-R})}$ . Vertical bars represent mean values.

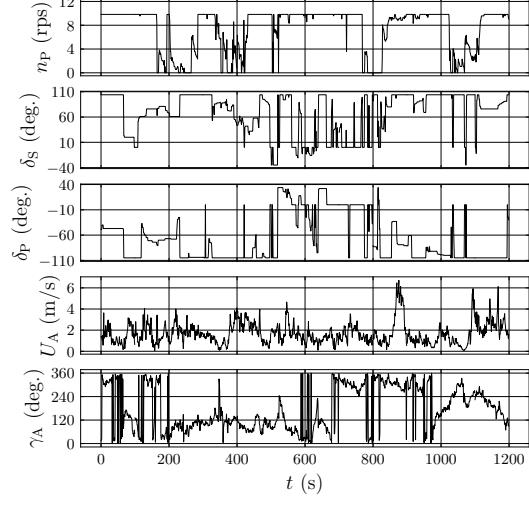
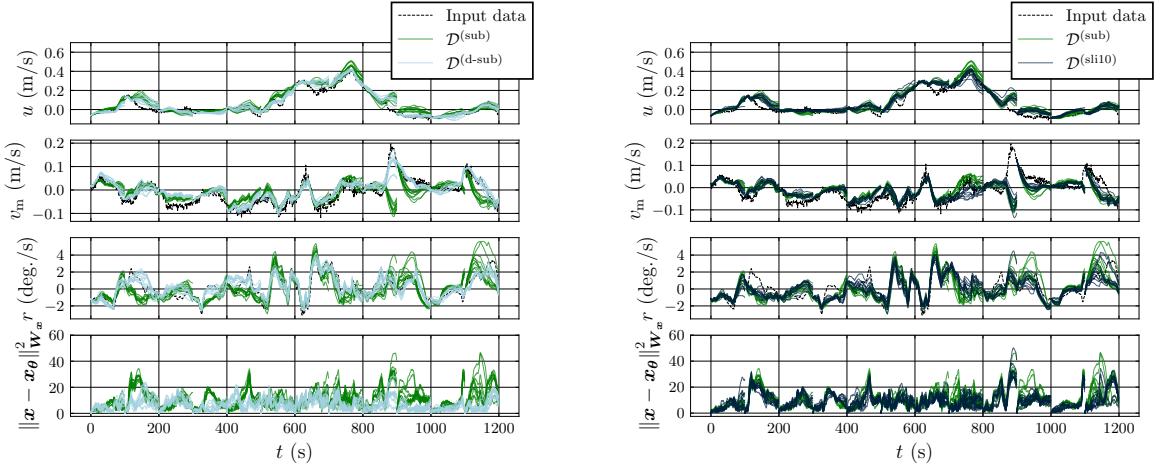
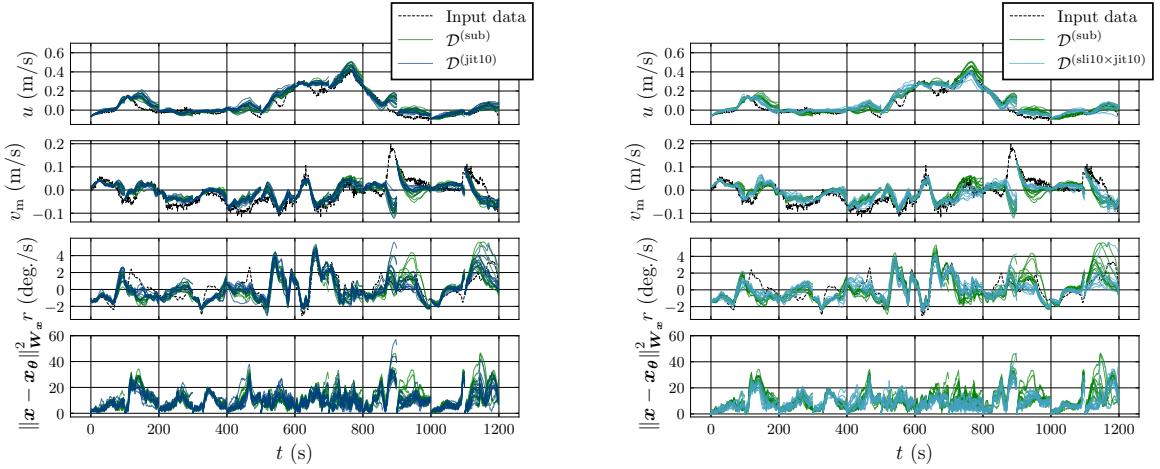


Fig. 4.9: Time series data of control inputs  $\mathbf{u}$  and apparent wind velocity and direction  $\boldsymbol{\omega}$  in random maneuver  $\mathcal{D}^{(\text{test-R})}$ .

$\mathcal{D}^{(\text{j1t10})}$  は  $\mathcal{D}^{(\text{d-sub})}$  より多くのデータ量を含んでいるにも関わらず、検証データ集合における推定誤差は小さくなっていない。この原因の一つとして、850 s 付近から 900 s にかけて発生している推定誤差の違いがある。Fig. 4.10aから、 $\mathcal{D}^{(\text{d-sub})}$  では推定誤差が小さくなっていることがわかるが、一方で、Figs. 4.10b–4.10dから、拡張されたデータセットでは推定誤差を減らせておらず、横速度の符号が逆転てしまっている。このとき、Fig. 4.9を見ると、 $\gamma_A = 300$  (deg.),  $U_A = 5.0$  (m/s) 程度の左前方からの比較的強い風が吹いていることがわかる。

ここで、Fig. 4.4から、 $\mathcal{D}^{(\text{sub})}$  と  $\mathcal{D}^{(\text{d-sub})}$  には、 $\gamma_A = 300$  (deg.),  $U_A = 5.0$  (m/s) に近い範囲のデータ量に大きく差が存在することがわかる。スライシングとジッタリングは、元のデータセットのどのデータにも近くないデータを合成することはできません。その結果、これらのデータ拡張方法では 850 秒から 900 秒に発生する予測誤差を低減することができなかったと考えられる。また、1100 秒から 1200 秒にかけて発生する予測誤差も、同様の理由によるものと考えられる。

最後に、船体の姿勢変数ベクトル  $\boldsymbol{\eta}$  の軌跡を Fig. 4.11 に示す。ここで、Fig. 4.11 では、700 秒から 1000 秒までのシミュレーション結果を示している。まず、Fig. 4.11a では、Surge 速度が比較的大きく、トラジェクトリは Yaw 速度の誤差の累積により乖離する傾向が見られる。特に、 $\mathcal{D}^{(\text{sub})}$  の乖離が大きく、 $\mathcal{D}^{(\text{d-sub})}$  と  $\mathcal{D}^{(\text{sli10xj1t10})}$  の乖離は比較的小さい。次に、Fig. 4.11b では、Surge 速度が大きく減少しており、比較的 850 s 付近から強い風が吹いている。 $\mathcal{D}^{(\text{d-sub})}$  の軌道は実験と比較的よく一致しているが、 $\mathcal{D}^{(\text{sub})}$  と  $\mathcal{D}^{(\text{sli10xj1t10})}$  の軌道は 850 s 以降で実験とは逆方向へ逸れてしまっている。そして、Fig. 4.11c では、Surge 速度がゼロに近く、Sway 方向に移動する運動が行われているところから始まっている。このケースでは  $\mathcal{D}^{(\text{d-sub})}$  が最も良い実験結果との一致を示しており、 $\mathcal{D}^{(\text{sli10xj1t10})}$  の軌道は  $\mathcal{D}^{(\text{sub})}$  よりも実験結果に近い。以上のことから、船体運動軌道においても、 $\mathcal{D}^{(\text{d-sub})}$  の船体運動軌道が実験結果と比較的良い一致を示していることがわかり、

(a) Results using dynamic models with optimal parameters trained by  $\mathcal{D}^{(\text{sub})}$  and  $\mathcal{D}^{(\text{d-sub})}$ .(b) Results using dynamic models with optimal parameters trained by  $\mathcal{D}^{(\text{sub})}$  and  $\mathcal{D}^{(\text{sli10})}$ .(c) Results using dynamic models with optimal parameters trained by  $\mathcal{D}^{(\text{sub})}$  and  $\mathcal{D}^{(\text{jit10})}$ .(d) Results using dynamic models with optimal parameters trained by  $\mathcal{D}^{(\text{sub})}$  and  $\mathcal{D}^{(\text{sli10}\times\text{jit10})}$ Fig. 4.10: Prediction result of ship state variables  $\nu$  and error  $d(\mathbf{x}^{\text{sim}}, \mathbf{x})$  in random maneuver  $\mathcal{D}^{(\text{test-R})}$ .

また、 $\mathcal{D}^{(\text{sub})}$  に比べて  $\mathcal{D}^{(\text{sli10}\times\text{jit10})}$  がより良く一致するケースが存在することがわかる。

最後に、着棧操船における獲得した操縦モデルの精度を示すため、検証データ集合 Fig. 4.12 に  $\mathcal{D}^{(\text{test-B})}$  の予測結果を示す。ただし、 $\mathcal{L}(\theta_{\text{opt}}; \mathcal{D}^{(\text{test})})$  の値が最も小さい最適パラメータを用いたモデルが使用された。Trajectory No.7 では、いずれの訓練データセットが用いられたモデルも縦速度  $u$  と横速度  $v_m$  は高い精度で予測されているが、回頭角速度  $r$  は、 $t = 30$  (s) 付近で、計測データと乖離してしまっている。そのため、Trajectory でも  $t = 30$  (s) 以降から乖離してしまっている。Trajectory No.8 では、 $\mathcal{D}^{(\text{sub})}$  で訓練されたモデルは、横速度  $v_m$  と回頭角速度  $r$  は高い精度で予測されているが、縦速度  $u$  は徐々に計測データから乖離してしまっている。そのため、船体運動軌道が計測データと大きく乖離してしまう結果となっている。一方で、

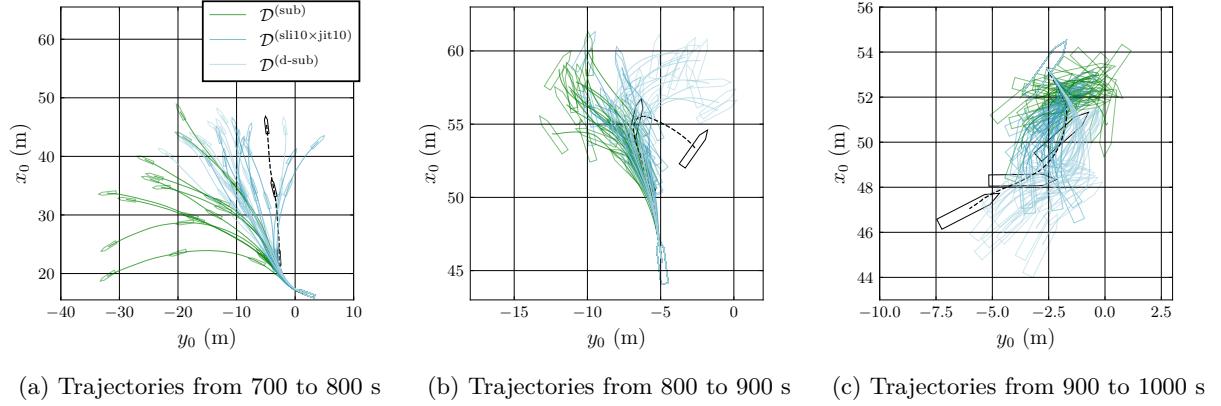


Fig. 4.11: Predicted results of trajectories in random maneuver  $\mathcal{D}^{(\text{test-R})}$ . Dynamic models with optimal parameters trained by  $\mathcal{D}^{(\text{sub})}$ ,  $\mathcal{D}^{(\text{d-sub})}$  and  $\mathcal{D}^{(\text{sli10}\times\text{jit10})}$  are used.

$\mathcal{D}^{(\text{d-sub})}$  と  $\mathcal{D}^{(\text{sli10}\times\text{jit10})}$  で訓練されたモデルは、縦速度  $u$  も高い精度で予測し続けられているため、船体運動軌道も計測データと近い挙動を示している。

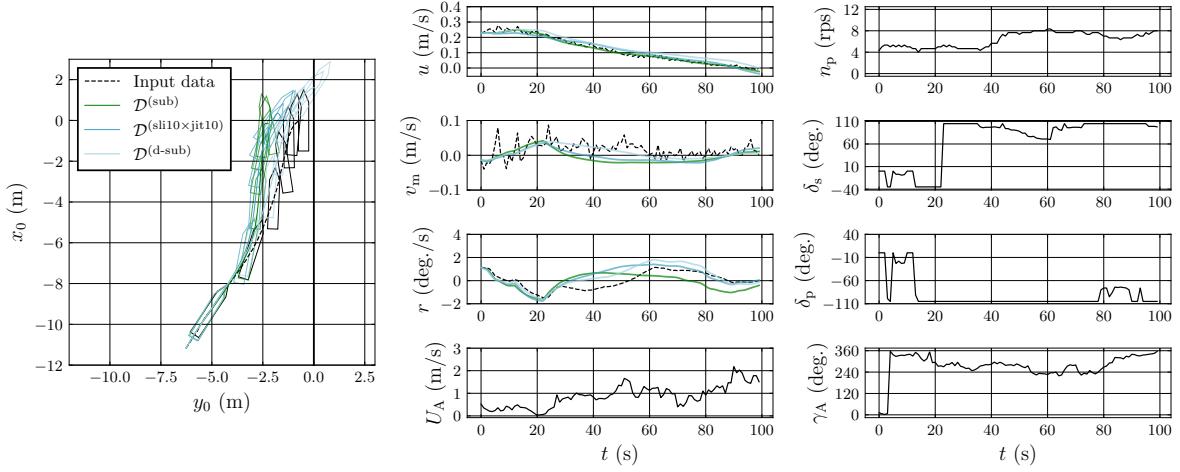
### 4.3 議論

4.2 節では、港内操船のための操縦モデルの同定問題におけるデータ拡張手法の比較実験の結果を示した。本実験では、FNN により操縦モデルが表現され、ランダム操船試験で計測された操縦運動データ集合が用いられた。実験結果から、スライシングやジッタリングにより検証データ集合に対する操縦モデルの予測精度が向上することが確認できた。また、スライシングとジッタリングは併用することでも予測精度を向上させることも示した。したがって、与えられる操縦運動データ集合が限られている場合には、スライシングとジッタリングは効果的なデータ補強法であることがわかった。

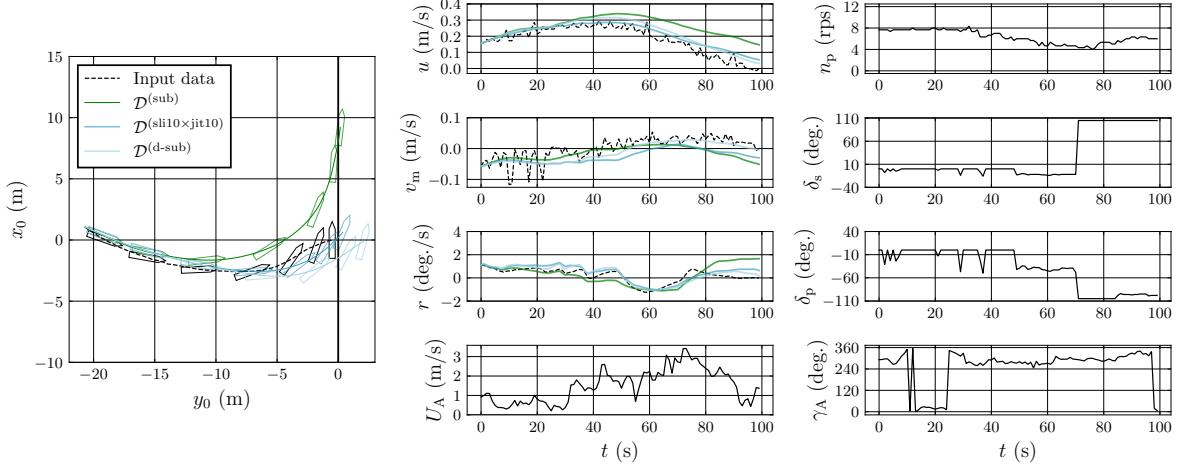
しかし、スライシングとジッタリングはデータセットのどのデータとも近接していないデータを合成することはできず、元のデータセットの外挿領域内で操縦モデルの予測精度を向上させることはできなかった。そのため、例えば訓練データには現れなかった強い風に遭遇すると予測精度が大きく低下する可能性がある。従って、ランダム操縦試験を用いる場合は、広く分布し、外挿領域の少ないデータを計測することが望ましい。ただし、実船試験では風などの外乱に関する状態量は制御できないため、計測時間が限られている場合、期待する望ましいデータを観測することは難しい。そのため、観測データの外挿状態の推定精度を向上させるためには、データから得られる情報に加えて物理的または流体力学的な知識を利用するアプローチも検討する必要があるかもしれない。

また、得られた操縦モデルは、着桟操船軌道を高精度に推定可能であることがわかった。これは、Fig. 4.2や Fig. 4.4を見てわかるように、ランダム操船の訓練データの分布が、着桟操船の分布をカバーしていることが一つの理由と考えられる。よって、ランダム操船によるデータ収集は港内操船のための操縦モデルの同定問題において有効な手段の一つであることがわかる。

しかし、ランダム操船試験は広範囲に分布する操縦データを収集可能であるが、手作業であるため、再現性に欠ける。従って、今後の研究では、データ収集方法に着目する必要があり、ジグザグ試験、旋回試験といっ



(a) Trajectory No. 7 of manual berthing maneuver. The diagonal hatch means an imaginary berth.



(b) Trajectory No. 8 of manual berthing maneuver. The diagonal hatch means an imaginary berth.

Fig. 4.12: Prediction results of manual berthing maneuver with an imaginary berth  $\mathcal{D}^{(\text{test-B})}$  using dynamic models with optimal parameters trained by  $\mathcal{D}^{(\text{sub})}$ ,  $\mathcal{D}^{(\text{d-sub})}$  and  $\mathcal{D}^{(\text{sli10}\times\text{jit10})}$ .

た従来の試験方法を取り入れ、再現性の高い試験手法の検討が必要であると考えられる。

## 4.4 結言

本章では、使用可能な操縦運動データ量が限られる場合に操縦モデルの推定精度を向上させることを目的に、ANN を用いた操縦モデルの推定にデータ拡張手法を導入し、模型船の自由航走試験データを利用した検証実験結果を示した。検証実験では、ランダム操船試験で計測された操縦運動データ集合を用いて、スライシングとジッタリングによって拡張された複数の訓練データ集合に対して操縦モデルの推定を行った。その結果、以下の点が判明した。

- 操縦運動データ量が限られている場合、スライシングとジッタリングにより操縦モデルの予測精度が改善された。
- スライシングとジッタリングは計測された操縦運動データに類似しないデータを合成することができないため、元のデータセットの外挿領域における操縦モデルの予測精度を改善することはできなかった。

そのため、本実験では、スライシングとジッタリングは操縦モデルの推定精度を向上させる可能性のある有効なデータ拡張手法であることが分かった。一方で、ランダム操船を離着桟制御のための操縦モデルの同定に用いる場合、予測精度を向上させるためにはデータの分布が広く分散していて、外挿が少ないデータを用意することが好ましいことがわかった。

## 第 5 章

# 強化学習を用いた軌道追従制御方策の獲得手法

本章では、船舶の軌道追従のための強化学習を用いた制御方策の獲得手法について詳述する。本手法では、モデルフリー強化学習アルゴリズムにより制御方策の訓練を実施するが、物理環境における試行錯誤を避けるため、船舶の操縦運動のシミュレーション環境をあらかじめ構築する。この操縦シミュレーション環境は、主に、操縦モデル、アクチュエータ応答モデル、風の確率過程モデルで構成される。

本節では、5.1 節で強化学習の概要を簡単に述べた後、5.2 節で軌道追従制御方策の獲得手法について述べる。また、5.3 節では訓練された軌道追従制御方策を未知の環境への適用方法について述べる。5.4 節及び5.5 節で提案手法の実験結果を示す。最後に、3.5 節では本章の結言について述べる。

### 5.1 強化学習の概要

まず、強化学習に関して簡単に説明する。強化学習 [88] は、エージェントが環境と相互作用し、試行錯誤を通じて累積報酬を最大化する行動方針を学習するフレームワークである。多くの強化学習アルゴリズムは環境の動的システムがマルコフ決定過程 (Markov Decision Process: MDP) としてモデル化されることを仮定している。MDP は以下の要素で構成される確率制御過程である。

- 状態集合  $\mathcal{S}$ : 環境がとり得る状態の集合を意味する。エージェントは環境からある離散時間ステップ  $t_k$  の  $s_k \in \mathcal{S}$  を観測する。
- 行動集合  $\mathcal{A}$ : エージェントが選択できる行動の集合を意味する。エージェントは、通常、観測された状態変数  $s_k$ に基づいて行動変数  $a_k \in \mathcal{A}$  を選択する。
- 状態遷移確率  $p(s_{k+1} | s_k, a_k)$ : エージェントが状態  $s_k$  で行動  $a_k$  を選択した場合、次に状態  $s_{k+1}$  に遷移する確率密度を表す。この状態遷移確率モデルは環境の動的システムによって決まる。
- 報酬関数  $r(s_{k+1}, s_k, a_k)$ : エージェントが状態  $s_k$  で行動  $a_k$  を選択したことで状態  $s_{k+1}$  に遷移した結果、得られる報酬を表す。
- $\gamma \in [0, 1]$ : 割引率は将来の報酬にどの程度の重みを与えるかを決定するパラメータを意味する。 $\gamma$  が 1 に近いほど、遠い将来の報酬を重視し、 $\gamma$  が 0 に近いほど現在の報酬を重視する。

本研究では、決定論的な制御方策関数を用いて行動変数が決定され、この制御方策関数を  $\mu$  と表記する。つ

まり、行動変数ベクトル  $\mathbf{a}$  は

$$\mathbf{a} = \mu(s) \quad (5.1.1)$$

と表される。状態変数ベクトル  $s$  の初期分布  $p(s_0)$  が与えられた時、制御方策関数  $\mu$  によって得られた一連の状態変数ベクトル  $s$  および行動変数ベクトル  $a$  を  $\tau = (s_0, a_0, \dots, s_{K-1}, a_{K-1}, s_K)$  と定義すると、一連の変数  $\tau$  が従う分布は

$$\rho_\mu = p(s_0) \prod_{k=1}^{K-1} p(s_{k+1} | s_k, \mu(s_k)) \quad (5.1.2)$$

と定義される。この時、強化学習の目的は、期待割引累積報酬

$$J_\mu = E_{\tau \sim \rho_\mu} \left[ \sum_{k=1}^{K-1} \gamma^{k-1} r(s_{k+1}, s_k, a_k) \right] \quad (5.1.3)$$

を最大化させる制御方策関数  $\mu$  を見つけることである。強化学習は、環境と相互作用を通してこの期待割引累積報酬  $J_\mu$  を最大化する最適制御方策関数  $\mu^*$  の探索をすることが可能である。

本研究では、最適制御方策関数  $\mu^*$  を見つけるため、アクター・クリティックベースの強化学習アルゴリズムである twin-delayed deep deterministic (TD3) policy gradient algorithm [102] を使用する。アクター・クリティックベースの強化学習アルゴリズムでは、制御方策関数  $\mu$  を表すアクターネットワークと価値関数を表すクリティックネットワークで構成される。クリティックネットワークは試行錯誤の経験を基にベルマン方程式の解である価値関数を学習し、アクターネットワークはクリティックネットワークを基に推定された方策勾配を用いてより多くの報酬を獲得するように改善される。

## 5.2 軌道追従制御方策の獲得手法

本節では、強化学習を用いた軌道追従制御のための制御方策関数  $\mu$  の獲得手法について詳述する。本研究では、時間  $t$  でパラメータ化された参照軌道が与えられると仮定する。与えられる参照軌道を  $\mathbf{x}^{(des)}(t)$  と表記すると、軌道追従制御の目的は与えられた参照軌道  $\mathbf{x}^{(des)}(t)$  に船を追従させることであり、この制御目的を実現させる制御方策関数  $\mu$  を強化学習を通して獲得する。ただし、物理環境での強化学習の試行錯誤を回避するため、訓練のための操縦シミュレーション環境をあらかじめ構築する。この操縦シミュレーション環境は操縦モデル、アクチュエータ応答モデル、風の確率過程モデルで構成されるが、操縦モデルは2.3.2 節及び2.4.2 節で述べられた MMG モデルを用いるか、もしくは、操縦運動データ集合があたえられ第 3 章で述べた ANN を用いたシステム同定アプローチなどを用いて推定される必要がある。

### 5.2.1 操縦運動のシミュレーション環境

本手法では、物理環境における試行錯誤を避けるため、制御方策関数  $\mu$  の訓練は操縦シミュレーション環境で実施される。この操縦シミュレーション環境は、操縦モデル、アクチュエータ応答モデル、風の確率過程モデルで構成される。以降では、それぞれのモデルおよびシミュレーション手法について詳述する。

#### 5.2.1.1 操縦モデル

船舶の操縦運動は Eq. (2.2.4) を数値的に解くことによってシミュレートすることが可能である。本論文では、Eq. (2.2.4b) は、水槽試験によって係数を求められた MMG モデル、もしくは、システム同定によって推定

された ANN モデルによって表現される。ただし、操縦モデルはマルコフ特性を持つと仮定され、Eq. (2.2.4b) の入力変数  $\mathbf{X}_{N_m}^{(-)}$ ,  $\mathbf{H}_{N_m}^{(-)}$ ,  $\mathbf{N}_{N_m}^{(-)}$ ,  $\mathbf{U}_{N_m}^{(-)}$ ,  $\mathbf{W}_{T,N_m}^{(-)}$ ,  $\mathbf{W}_{A,N_m}^{(-)}$  はすべて  $\mathbf{x}$ ,  $\boldsymbol{\eta}$ ,  $\boldsymbol{\nu}$ ,  $\mathbf{u}$ ,  $\mathbf{w}_T$ ,  $\mathbf{w}_A$  に置き換わる。

### 5.2.1.2 アクチュエータ応答モデル

バウスラスターの回転数や舵角は急激に変化させることはできず、指示値に対して遅れが発生する。本手法では、実際の応答特性を考慮して制御方策関数  $\mu$  の訓練を実施するため、アクチュエータの応答特性が表すモデルが操縦シミュレーション環境へ組み込まれている。本研究では、供試船の応答特性を考慮した結果、舵角、プロペラ回転数、バウスラスター回転数の時間変化率の大きさが一定であると仮定した。つまり、アクチュエータの状態変数  $\mathbf{u}$  のある一つの成分を  $y$  とし、その指示値を  $r$  と表記すると、応答特性はスロープを持つ階段関数でモデル化され、

$$\dot{y} = \begin{cases} K_y^{(\text{act})} & \text{for } \epsilon \leq y \\ K_y^{(\text{act})} (r - y) / \epsilon & \text{for } -\epsilon < r - y < \epsilon \\ -K_y^{(\text{act})} & \text{for } y \leq -\epsilon \end{cases} \quad (5.2.1)$$

と表される。ここで、 $K_y^{(\text{act})}$  は変数  $y$  の時間変化率の大きさを表す定数で、 $\epsilon$  は定数である。ここで、スロープは数値シミュレーションにおけるアクチュエータ状態変数の振動を回避するために導入されており、 $\epsilon$  は数値シミュレーションのタイムステップに等しい値が使用される。

したがって、制御方策関数  $\mu$  はアクチュエータ状態変数ベクトル  $\mathbf{u} \in \mathbb{R}^{N_u}$  を直接選択するのではなく、その指示値を選択することができる。簡単のために、制御方策関数  $\mu$  の行動変数を  $\mathbf{a} \in \mathbb{R}^{N_u}$  と表記すると、考慮される応答特性をまとめて

$$\dot{\mathbf{u}} = \mathbf{f}^{(\text{act})} (\mathbf{u}, \mathbf{a}) \quad (5.2.2)$$

と表記する。

### 5.2.1.3 風外乱の確率過程モデル

風は、通常、不規則に変動するため、制御方策関数  $\mu$  は様々な風状態に対して最適な行動を選択する必要がある。そのため、訓練では変動する風を考慮する。風の過程は牧ら [127, 128] によって提案された方法を用いて生成される。この方法では、風速および風向の過程は1次元フィルタ方程式

$$\begin{cases} dU_T = \alpha_U (U_T - \bar{U}_T) dt + \sigma_U dW \\ d\xi_T = \alpha_\xi (\xi_T - \bar{\xi}_T) dt + \sigma_\xi dW \end{cases} \quad (5.2.3)$$

で表されると仮定される。ここで、 $\alpha_U < 0, \alpha_\xi < 0, \sigma_U, \sigma_\xi$  はフィルター係数、 $\bar{U}_T$  および  $\bar{\xi}_T$  はそれぞれ風速および風向の平均値、 $dW$  はウィーナー過程の増加分である。風速に関するフィルター係数は日野スペクトラムで近似された風速スペクトルを用いて決定され、風向に関するフィルター係数は桑島らが提案した平均風速に対する風向変動の標準偏差の回帰式を用いて決定される。フィルター係数の詳細は文献 [127, 128] を参照されたい。

### 5.2.1.4 離散化システムノイズ

連続時間モデルの離散化手法について説明する。本研究では、操縦モデルおよびアクチュエータ応答モデルは前進オイラー法により離散化され、風外乱の確率過程モデルは、オイラー・丸山法によって離散化される。

離散化された差分方程式は

$$\begin{cases} \boldsymbol{\eta}(t + \Delta t) = \boldsymbol{\eta}(t) + \Delta t \cdot \mathbf{R}(\boldsymbol{\eta}(t)) \boldsymbol{v}(t) \\ \boldsymbol{v}(t + \Delta t) = \boldsymbol{v}(t) + \Delta t \cdot \mathbf{F}'(\boldsymbol{v}(t), \boldsymbol{u}(t), \boldsymbol{w}_T(t)) + \boldsymbol{\epsilon}^{(sys)} \\ \boldsymbol{u}(t + \Delta t) = \boldsymbol{u}(t) + \Delta t \cdot \mathbf{f}^{(act)}(\boldsymbol{u}(t), \boldsymbol{a}(t)) \\ U_T(t + \Delta t) = U_T(t) + \Delta t \cdot \alpha_U(U_T(t) - \bar{U}_T) + \sigma_U \Delta W \\ \xi_T(t + \Delta t) = \xi_T(t) + \Delta t \cdot \alpha_\xi (\xi_T(t) - \bar{\xi}_T) + \sigma_\xi \Delta W \end{cases} \quad (5.2.4)$$

と定義される。ここで、 $\boldsymbol{\epsilon}^{(sys)} \sim \mathcal{N}\left(\mathbf{0}, \text{diag}\left(\left(\sigma_u^{(sys)}\right)^2, \left(\sigma_{v_m}^{(sys)}\right)^2, \left(\sigma_r^{(sys)}\right)^2\right)\right)$  で、 $\sigma_u^{(sys)}, \sigma_{v_m}^{(sys)}, \sigma_r^{(sys)}$  はシステムノイズの標準偏差、 $\Delta W$  はウィーナー過程の増加分を表し、 $\xi \sim \mathcal{N}(0, 1)$  とすると  $\Delta W = \sqrt{\Delta t} \xi$  である。また、 $\mathbf{F}'$  は Eq. (2.2.4b) と Eq. (2.2.2) の合成関数であることに注意する。

### 5.2.1.5 観測ノイズ

エージェントは操縦シミュレーション環境から船体の運動学的変数ベクトル  $\boldsymbol{x}$ 、アクチュエータ状態変数ベクトル  $\boldsymbol{u}$ 、真風速風向ベクトル  $\boldsymbol{w}_T$  を観測し、行動変数ベクトル  $\boldsymbol{a}$  を決定する。物理環境では、GNSS やアネモメータなどの計測機器を通してこれらの変数が観測されるため、観測値には観測誤差が含まれる可能性が高い。そこで、物理環境でとり得る誤差を操縦シミュレーション環境で再現するため、船体の運動学的変数ベクトル  $\boldsymbol{x}$  に関して観測ノイズを考える。具体的には、船体の運動学的変数ベクトル  $\boldsymbol{x}$  の観測値は  $\boldsymbol{\epsilon}^{(obs)} \sim \mathcal{N}\left(\mathbf{0}, \text{diag}\left(\left(\sigma_{x_0}^{(obs)}\right)^2, \left(\sigma_{y_0}^{(obs)}\right)^2, \left(\sigma_\psi^{(obs)}\right)^2, \left(\sigma_u^{(obs)}\right)^2, \left(\sigma_{v_m}^{(obs)}\right)^2, \left(\sigma_r^{(obs)}\right)^2\right)\right)$  で表される観測ノイズが付加される。

## 5.2.2 軌道追従制御問題

軌道追従制御の目的は与えられた参照軌道  $\boldsymbol{x}^{(des)}(t)$  に船を追従させることであり、この制御目的を実現させる制御方策関数  $\boldsymbol{\mu}$  を強化学習を通して獲得する。制御方策関数  $\boldsymbol{\mu}$  は参照軌道  $\boldsymbol{x}^{(des)}(t)$  および観測可能な船体の運動学的変数ベクトル  $\boldsymbol{x}$ 、アクチュエータ状態変数ベクトル  $\boldsymbol{u}$ 、真風速風向ベクトル  $\boldsymbol{w}_T$  に関する情報を基に行動変数を決定する。本手法では、軌道追従制御をあらかじめ決められた条件を満たすまでのエピソディックタスクとして定式化され、訓練時に与えられる参照軌道  $\boldsymbol{x}^{(des)}(t)$  は自動的に生成される。以降、5.2.2.1 節で制御方策関数  $\boldsymbol{\mu}$  の状態変数について、5.2.2.3 節で報酬関数  $r$  とエピソードの終了条件について、5.2.2.2 節では、訓練時における参照軌道の自動生成手法について詳述する。

### 5.2.2.1 観測状態変数

エージェントが観測する状態変数について詳述する。エージェントは与えられた参照軌道  $\boldsymbol{x}^{(des)}(t)$  と操縦シミュレーション環境から現在時刻で観測された船体の運動学的変数ベクトル  $\boldsymbol{x}$ 、アクチュエータ状態変数ベクトル  $\boldsymbol{u}$ 、真風速風向ベクトル  $\boldsymbol{w}_T$  を入力として受け取ることが可能である。軌道追従制御を実現させるためには、制御方策関数  $\boldsymbol{\mu}$  への入力変数をこれらの情報から適切に選択する必要がある。以降では、制御方策関数  $\boldsymbol{\mu}$  へ与える状態変数について詳述する。

#### (1) 追従誤差

まず、追従誤差に関する変数について述べる。参照軌道  $\boldsymbol{x}^{(des)}(t)$  は時刻  $t$  の関数であり、全時刻における参照軌道  $\boldsymbol{x}^{(des)}(t)$  を直接制御方策関数  $\boldsymbol{\mu}$  へ入力することは難しい。また、参照軌道  $\boldsymbol{x}^{(des)}(t)$  の姿勢変数成

分を  $\boldsymbol{\eta}^{(\text{des})}(t) \in \mathbb{R}^3$  と表記すると、参照姿勢変数ベクトル  $\boldsymbol{\eta}^{(\text{des})}$  および船体の姿勢変数ベクトル  $\boldsymbol{\eta}$  は空間固定座標  $O_0 - x_0y_0z_0$  で定義されているため、それぞれの絶対的な位置および角度に関する情報に意味はない。そのため、参照軌道  $\mathbf{x}^{(\text{des})}(t)$  を直接制御方策関数  $\boldsymbol{\mu}$  へ入力されない。

本研究では、軌道追従制御の制御目的は現在および未来の追従誤差を最小化することであると言え換えることができる。そのため、ある時刻  $t$  の船体の運動学的変数ベクトル  $\mathbf{x}$  と現在時刻  $t$  から  $T$  時間先の参照軌道  $\mathbf{x}^{(\text{des})}(t)$  の誤差へ変換された後、制御方策関数  $\boldsymbol{\mu}$  へ入力される。ただし、本研究では、船体の速度変数に関する成分の追従誤差は使用されず、代わりに将来時刻の船体の姿勢変数に関する追従誤差が与えられる。具体的には、参照軌道  $\mathbf{x}^{(\text{des})}(t)$  の姿勢変数成分を  $\boldsymbol{\eta}^{(\text{des})}(t) \in \mathbb{R}^3$  と表記すると、ある時刻  $t$  の船体の姿勢変数ベクトル  $\boldsymbol{\eta}$  と現在時刻  $t$  から  $T$  時間先の参照姿勢変数ベクトル  $\boldsymbol{\eta}^{(\text{des})}$  の誤差は

$$\mathbf{e}_T^{(\text{des})}(t) = \mathbf{R}^{-1}(\boldsymbol{\eta}(t)) (\boldsymbol{\eta}^{(\text{des})}(t+T) - \boldsymbol{\eta}(t)) \quad (5.2.5)$$

と定義される。このとき、追従誤差に関する変数は

$$\mathbf{s}^{(\text{des})} = \left( \left( \mathbf{e}_0^{(\text{des})} \right)^T, \left( \mathbf{e}_{T_1}^{(\text{des})} \right)^T, \left( \mathbf{e}_{T_2}^{(\text{des})} \right)^T, \dots, \left( \mathbf{e}_{T_N}^{(\text{des})} \right)^T \right)^T \in \mathbb{R}^{(3(1+N))} \quad (5.2.6)$$

と定義される。ここで、 $T_1, T_2, \dots, T_N$  は制御方策関数  $\boldsymbol{\mu}$  に与える参照状態の将来時刻を表す。従って、制御方策関数  $\boldsymbol{\mu}$  には有限個の将来時刻点における参照状態との追従誤差が与えられる。

## (2) その他の状態変数

船舶の操縦運動は Eq. (2.2.7) で定義されるように船体の運動学的変数ベクトル  $\mathbf{x}$ 、アクチュエータ状態変数ベクトル  $\mathbf{u}$  および真風速風向ベクトル  $\mathbf{w}_T$  に影響を受ける。参照軌道  $\mathbf{x}^{(\text{des})}(t)$  および船体の姿勢変数ベクトル  $\boldsymbol{\eta}$  に関連した状態変数は Eq. (5.2.6) で定義されるが、その他の船舶の操縦運動に影響を与える状態変数についても制御方策関数  $\boldsymbol{\mu}$  へ入力される必要がある。従って、本研究では、船体の速度変数ベクトル  $\mathbf{v}$ 、アクチュエータ状態変数ベクトル  $\mathbf{u}$  および相対風速風向ベクトル  $\mathbf{w}_A$  が制御方策関数  $\boldsymbol{\mu}$  へ入力される。

### 5.2.2.2 参照軌道生成

軌道追従制御のための制御方策関数  $\boldsymbol{\mu}$  を訓練するには参照姿勢変数ベクトル  $\boldsymbol{\eta}^{(\text{des})}$  が与えられる必要があるが、与えられる参照姿勢変数ベクトル  $\boldsymbol{\eta}^{(\text{des})}$  は目的に応じて決定される必要がある。これは、エージェントに追従させたい軌道を追従する経験を訓練で積ませることで、目的の制御方策関数  $\boldsymbol{\mu}$  を獲得できるためである。

本研究の目的は港内操船の自動化であるため、訓練では港内操船でとり得る軌道を参照軌道として与えられる必要がある。対象港湾における港内操船のオペレーションデータを訓練時の参照軌道として使用可能であるかもしれないが、必ずしも十分なデータを用意できるわけではなく、少数の特定の軌道に基づく学習は望ましくない。そのため、低速操縦運動を含む港内操船でとり得るさまざまな運動軌道を与えるため、訓練では以下の軌道を与える：

- 定点保持軌道:  $\mathbf{x}^{(\text{des})}(t) = (0, 0, 0, 0, 0, 0)^T$
- ランダムに制御入力が決定された操縦運動軌道: 追従可能な参照姿勢変数ベクトル  $\boldsymbol{\eta}^{(\text{des})}$  を生成するため、5.2.1 節で述べた操縦シミュレーション環境を用いて、操縦シミュレーションを行う。操縦シミュ

レーションでは、真風速風向は平均風速風向に固定され、行動変数ベクトル  $\mathbf{a}$  は

$$\mathbf{a}(t) = \sum_{k=0}^{K-1} \mathbf{1}_{t \in [t_k, t_{k+1})} \mathbf{a}_k^{(\text{rand})} \quad (5.2.7)$$

で与えられた。ここで  $\mathbf{a}_k^{(\text{rand})}$  はランダムに決定される変数である。また、船体の運動学的変数ベクトル  $\mathbf{x}$ 、アクチュエータ状態変数ベクトル  $\mathbf{u}$  および真風速風向ベクトル  $\mathbf{w}_T$  の初期値は、目的の港内操船で取りうる状態変数を中心にランダムに選択される。もししくは、与えられた操縦運動データ集合の中からランダムに選択される。

ただし、港内操船でとり得るさまざまな運動軌道を生成するためには、 $\mathbf{a}_k^{(\text{rand})}$  をアクチュエータの種類に応じて適切に決定する必要がある。本論文では、5.4 節及び5.5 節で VecTwin 舵システムとバウスラスタが搭載されている供試船 2 を用いた検証実験の結果を示す。そのため、供試船 2 のための  $\mathbf{a}_k^{(\text{rand})}$  の決定方法例について述べる。

### (1) 供試船 2 のためのランダム制御入力の決定方法例

供試船 2 には、2.4 節で述べたように、VecTwin 舵システムとバウスラスタが搭載されている。VecTwin 舵システムは左右の舵角に応じてさまざまな推力を発生させられるが、プロペラと舵の推力が釣り合い、全体的な合力が見かけ上ゼロになる舵角が存在する。この舵角はホバー角と呼ばれ、この舵角は  $(\delta_P, \delta_S) \approx (-75^\circ, 75^\circ)$  である。よって、VecTwin 舵システムで低速を中心とした操縦運動を実現するためには、このホバー角を中心に左右の舵角を分布させる必要がある。

そのため、5.4 節に述べる検証実験では  $\mathbf{a}_k^{(\text{rand})}$  の  $\delta_P$  成分は  $\mathcal{N}(-75^\circ, (30^\circ)^2)$ 、 $\delta_S$  成分は  $\mathcal{N}(75^\circ, (30^\circ)^2)$  に従って生成され、 $n_{BT}$  成分は  $[-30, 30]$  における一様分布に従って生成された。

また、5.5 節に述べる検証実験では、 $\mathbf{a}_k^{(\text{rand})}$  は多変量ガウス分布  $\mathcal{N}(\boldsymbol{\mu}^{(\text{act})}, \boldsymbol{\Sigma}^{(\text{act})})$  に従って生成され、平均ベクトル  $\boldsymbol{\mu}^{(\text{act})}$  および共分散行列  $\boldsymbol{\Sigma}^{(\text{act})}$  は

$$\boldsymbol{\mu}^{(\text{act})} = \begin{pmatrix} -80 \\ 80 \\ 0.0 \end{pmatrix}, \quad \boldsymbol{\Sigma}^{(\text{act})} = \begin{bmatrix} 30^2 & 0.0 & 0.0 \\ 0.0 & 30^2 & 0.0 \\ 0.0 & 0.0 & 15^2 \end{bmatrix} \quad (5.2.8)$$

とされた。実験では、プロペラ回転数は  $n_P = 10.0$  (rps) で固定されていることに注意する。

港内操船は前進速度が十分大きい標準的な操縦運動も含まれるため、左右の舵角がゼロに近い領域で分布する参照軌道の生成もまた生成される必要がある。そのため、5.5 節に述べる検証実験では、Eq. (5.2.8) で定義される平均ベクトル  $\boldsymbol{\mu}^{(\text{act})}$  および共分散行列  $\boldsymbol{\Sigma}^{(\text{act})}$  の値の他に

$$\boldsymbol{\mu}^{(\text{act})} = \begin{pmatrix} -40 \\ 40 \\ 0.0 \end{pmatrix}, \quad \boldsymbol{\Sigma}^{(\text{act})} = \begin{bmatrix} 30^2 & -0.3 \times 30^2 & 0 \\ -0.3 \times 30^2 & 30^2 & 0 \\ 0 & 0 & 10^2 \end{bmatrix} \quad (5.2.9)$$

および

$$\boldsymbol{\mu}^{(\text{act})} = \begin{pmatrix} 0.0 \\ 0.0 \\ 0.0 \end{pmatrix}, \quad \boldsymbol{\Sigma}^{(\text{act})} = \begin{bmatrix} 15^2 & 0.75 \times 15^2 & 0.0 \\ 0.75 \times 15^2 & 15^2 & 0.0 \\ 0.0 & 0.0 & 1 \end{bmatrix} \quad (5.2.10)$$

が用いられた。

### 5.2.2.3 報酬関数と終了条件

次に、報酬関数  $r$  とエピソードの終了条件について説明する。すでに述べたように、強化学習の目的は追従誤差を最小化する制御方策関数  $\mu$  を獲得することである。よって、報酬関数  $r$  は追従誤差が小さいほど大きな報酬が与えられるように設計される必要がある。本研究では、報酬関数  $r$  は正の定数から追従誤差ペナルティを引くことで定義される。検証実験で用いられた詳細な定義はそれぞれ5.4.1節及び5.5.2節で示すが、ここでは、使用される追従誤差ペナルティについて述べる。また、実用的に発生する可能性の高い状態(追従誤差)を中心に訓練を実施するため、エピソードを終了させる追従誤差の閾値を設定する。

#### (1) 追従誤差ペナルティ

まず、追従誤差に関する報酬のペナルティについて述べる。追従姿勢誤差ベクトル  $e_0^{(\text{des})}$  は Eq. (5.2.5) で定義されるが、報酬のペナルティとするにはスカラーへ変換される必要がある。この変換方法として、L2ノルム  $\|e_0^{(\text{des})}\|$  やある行列  $W \in \mathbb{R}^{3 \times 3}$  で重み付けされた L2ノルム  $\|e_0^{(\text{des})}\|_W$  などが考えられる。座標と回頭角は単位が異なるため、重みの調整が可能な重み付き L2ノルムがより適切ではあるものの、適切な重みの調整には試行錯誤が必要である。特に、回頭角誤差を過剰に重視するように重みが設定されている場合、一時的に回頭角誤差を大きく取ることで位置座標誤差が効率的に減少させる手段が選ばれなくなる可能性があり、結果として位置座標誤差の減少が遅れてしまう可能性がある。

そこで、本研究では、船体長さに応じた追従誤差を表すスカラー指標として、Fig. 5.1に示すような船首および船尾位置におけるユークリッド距離誤差を導入する。ある船体の姿勢変数ベクトル  $\eta$  に対する船首船尾位置はそれぞれ

$$\begin{cases} p^{(\text{bow})}(\eta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \left( \eta + R(\eta) \begin{pmatrix} L_{\text{ship}}/2 \\ 0 \\ 0 \end{pmatrix} \right) \\ p^{(\text{stn})}(\eta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \left( \eta + R(\eta) \begin{pmatrix} L_{\text{ship}}/2 \\ 0 \\ 0 \end{pmatrix} \right) \end{cases} \quad (5.2.11)$$

と表される。ここで、 $L_{\text{ship}}$  は船体長さを表す。この時、船首と船尾におけるユークリッド距離誤差はそれぞれ

$$\begin{cases} e^{(\text{bow})}(t) = \|p^{(\text{bow})}(\eta^{(\text{des})}(t)) - p^{(\text{bow})}(\eta(t))\| \\ e^{(\text{stn})}(t) = \|p^{(\text{stn})}(\eta^{(\text{des})}(t)) - p^{(\text{stn})}(\eta(t))\| \end{cases} \quad (5.2.12)$$

と定義される。本研究ではこれらのユークリッド距離誤差が追従誤差ペナルティとして使用される。

$e^{(\text{bow})}$  および  $e^{(\text{stn})}$  の  $e_0^{(\text{des})}$  に関する変動を示す。 $e_0^{(\text{des})}$  の要素をそれぞれ  $\hat{x}, \hat{y}, \hat{\psi}$  と表記したとき、位置座標誤差  $\sqrt{\hat{x}^2 + \hat{y}^2}/L_{\text{ship}}$  と回頭角誤差  $\hat{\psi}$  に関する  $e^{(\text{bow})}$  および  $e^{(\text{stn})}$  の平均値の変化を Fig. 5.2に示す。この結果から、位置座標誤差が船体長さに対して十分大きい場合、回頭角誤差に対して  $e^{(\text{bow})}$  および  $e^{(\text{stn})}$  の平均値は大きく変化しないことがわかる。一方、位置座標誤差が船体長さに対して小さい場合、 $e^{(\text{bow})}$  および  $e^{(\text{stn})}$  の平均値は回頭角誤差に大きく依存することが分かる。

#### (2) 終了条件

エピソードを終了させる追従誤差の閾値について述べる。この閾値は一定以上の追従誤差を持つ状態における訓練を除外させることができ、エージェントが遭遇する可能性の高い状態を中心に訓練を実施するため、こ

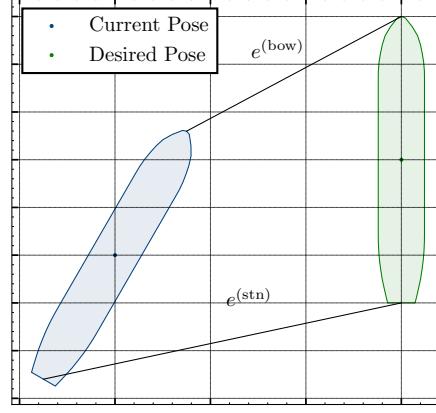
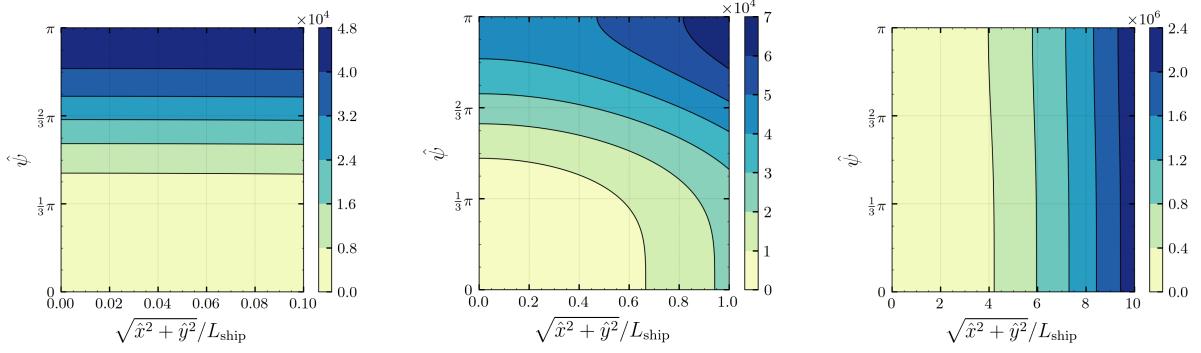


Fig. 5.1: Euclidean distance error at the bow and stern positions.



(a)  $0 \leq \sqrt{\hat{x}^2 + \hat{y}^2}/L_{\text{ship}} \leq 0.1$ : The area where the positional error is small in relation to ship length.

(b)  $0 \leq \sqrt{\hat{x}^2 + \hat{y}^2}/L_{\text{ship}} \leq 1$ : The area where the positional error is close to ship length.

(c)  $0 \leq \sqrt{\hat{x}^2 + \hat{y}^2}/L_{\text{ship}} \leq 10$ : The area where the positional error is large in relation to ship length.

Fig. 5.2: Mean Euclidean distance of tracking errors at bow and stern positions  $(e^{(\text{bow})} + e^{(\text{stn})})/2$ .

の閾値が設定される。また、より多くの累積報酬の獲得のため、より長い時間の間、常に追従誤差を小さく保つような制御方策関数  $\mu$  を獲得するため、閾値はエピソード時間に応じて減少するように設計される。よって、 $e^{(\text{bow})}$  および  $e^{(\text{stn})}$  の閾値を  $e^{(\text{tol})}$  と表記すると、閾値  $e^{(\text{tol})}$  はエピソード時間  $t$  を用いて

$$e^{(\text{tol})}(t) = \left( e_0^{(\text{tol})} - e_\infty^{(\text{tol})} \right) \exp(-\lambda_e t) + e_\infty^{(\text{tol})} \quad (5.2.13)$$

と定義される。ここで、 $e_0^{(\text{tol})}$ ,  $e_\infty^{(\text{tol})}$ ,  $\lambda_e$  はそれぞれ、上限値の初期値、終端値、時間減少率を表す定数である。従って、 $e^{(\text{bow})}$  および  $e^{(\text{stn})}$  が閾値  $e^{(\text{tol})}$  より大きくなると、エピソードが終了され、それ以降の時刻では報酬が与えられない。

#### 5.2.2.4 アクターネットワーク構造

本研究では TD3 を用いて最適制御方策関数  $\mu^*$  の探索が行われる。この手法では、制御方策関数  $\mu$  が順伝播型ニューラルネットワークによって表現されるが、この順伝播型ニューラルネットワークは全ての行動変数

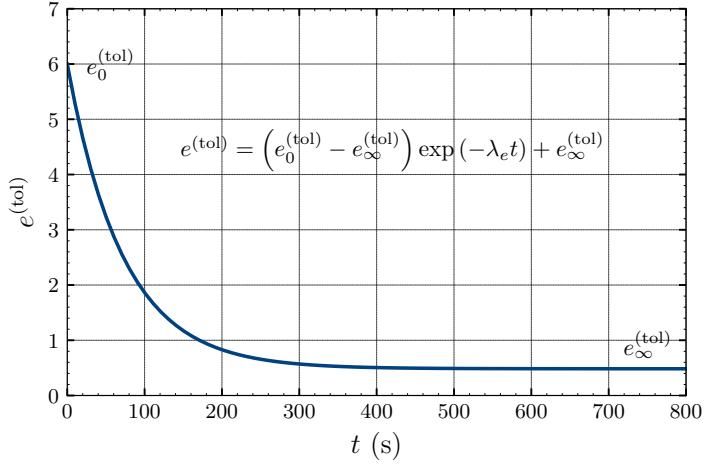


Fig. 5.3: Threshold for tracking error that decreases with episode time.

$a \in \mathcal{U}$  を表現可能である必要がある。そのため、 $N_u$  次元へ変換された出力層の活性化関数にはシグモイド関数が用いられ、その後、

$$a_i = (\bar{u}_i - \underline{u}_i) y_i + \underline{u}_i \text{ for } i = 1, \dots, N_u \quad (5.2.14)$$

により線形変換される。ここで、 $a_i$  および  $y_i$  はそれぞれ  $i$  番目成分の行動変数および順伝播型ニューラルネットワークの出力変数である。

### 5.2.3 静的障害物の考慮

港湾内での船舶運航は、通常の航行に比べて複雑な環境であり、周囲には静的な障害物が多数存在する。特に、離着桟操船では、港内の岸壁や桟橋、係留設備などの固定された障害物との距離が小さくなるが、環境外乱下でそれらの障害物との衝突を避けるように制御されることが必要不可欠である。通常、軌道追従制御のために与えられる参照軌道  $\mathbf{x}^{(des)}(t)$  には空間制約が考慮されているため、少なくとも静的障害物との衝突は発生しない。しかし、離着桟操船のように障害物との距離が小さい操船のための参照軌道  $\mathbf{x}^{(des)}(t)$  が与えられた場合、少しの追従誤差が障害物との衝突を引き起こす可能性がある。従って、船舶が目標地点へ安全に到達するためには、軌道追従制御でもまた障害物を考慮した上で適切に操船制御される必要がある。具体的には、実環境ではモデル化誤差や未知外乱影響により追従誤差を無くすることは不可能であるが、少なくとも追従誤差により引き起こされる障害物との衝突を回避することが求められる。そこで、本研究では、モデル化誤差や未知外乱の影響により、追従誤差の発生を避けられない場合であったとしても、障害物との衝突の可能性の高くなる追従誤差は避ける軌道追従制御のための制御方策関数  $\mu$  を獲得手法を提案する。

ここで、静的障害物の位置情報は観測可能であると仮定され、静的障害物の形状は  $x_0y_0$  平面における複数の多角形で表現されるとする。このとき、ある多角形領域集合は

$$P = \left\{ \mathbf{p} \in \mathbb{R}^2 \mid \mathbf{p} = \lambda_1 \mathbf{v}_1 + \lambda_2 \mathbf{v}_2 + \cdots + \lambda_{N_v} \mathbf{v}_{N_v}, \quad \sum_{i=1}^{N_v} \lambda_i = 1, \quad \mathbf{v}_i \in \mathbb{R}^2, \quad \lambda_i \geq 0 \text{ for } i = 1, \dots, N_v \right\} \quad (5.2.15)$$

と定義される。ここで、 $v_i \in \mathbb{R}^2$  は多角形のある頂点である。そして、全ての障害物領域は

$$\mathcal{O} = \bigcup_{i=1}^{N_P} P_i \quad (5.2.16)$$

と定義される。ここで、 $P_i$  はある多角形領域集合を表し、 $N_P$  は多角形の総数を表す。この静的障害物領域集合  $\mathcal{O}$  があらかじめ与えられるものと仮定される。

### 5.2.3.1 観測状態変数

静的障害物に関して、エージェントが観測する変数について詳述する。エージェントは静的障害物領域集合  $\mathcal{O}$  が観測可能であると仮定されるが、静的障害物領域集合  $\mathcal{O}$  は次元が大きいため、制御方策関数  $\mu$  の入力変数には適していない。そこで、参照姿勢変数ベクトル  $\eta^{(des)}$  から最も近い障害物の位置座標と船体位置座標の誤差を静的障害物領域集合  $\mathcal{O}$  と船体との位置関係に関する特徴量として用いる。ここで、ある時刻  $t$  において参照姿勢変数ベクトル  $\eta^{(des)}$  から最も近い障害物の点は

$$\mathbf{o}^{(\text{near})}(t) = \underset{\mathbf{o} \in \mathcal{O}}{\operatorname{argmin}} \left\| \mathbf{o} - \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \eta^{(\text{des})}(t) \right\| \quad (5.2.17)$$

と定義される。このとき、時刻  $t$  の  $T$  時間先の参照姿勢変数ベクトル  $\eta^{(\text{des})}$  から最も近い障害物点と船体位置の誤差は

$$\mathbf{e}_T^{(\text{obs})}(t) = \begin{bmatrix} \cos \psi(t) & \sin \psi(t) \\ -\sin \psi(t) & \cos \psi(t) \end{bmatrix} \left( \mathbf{o}^{(\text{near})}(t+T) - \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \eta(t) \right) \quad (5.2.18)$$

と定義される。よって、静的障害物領域集合  $\mathcal{O}$  に関する制御方策関数  $\mu$  の入力変数を

$$\mathbf{s}^{(\text{obs})} = \left( (\mathbf{e}_0^{(\text{obs})})^\top, (\mathbf{e}_{T_1}^{(\text{obs})})^\top, (\mathbf{e}_{T_2}^{(\text{obs})})^\top, \dots, (\mathbf{e}_{T_N}^{(\text{obs})})^\top \right)^\top \in \mathbb{R}^{2(1+N)} \quad (5.2.19)$$

と定義する。ここで、 $T_1, T_2, \dots, T_N$  は参照姿勢変数ベクトル  $\eta^{(\text{des})}$  の将来時刻である。

### 5.2.3.2 静的障害物生成

制御方策関数  $\mu$  を訓練するためには、参照姿勢変数ベクトル  $\eta^{(\text{des})}$  と同様に静的障害物領域集合  $\mathcal{O}$  が与えられる必要がある。ただし、参照姿勢変数ベクトル  $\eta^{(\text{des})}$  は5.2.2.2 節で述べた手法によって自動的に生成されるため、静的障害物領域集合  $\mathcal{O}$  は生成された参照姿勢変数ベクトル  $\eta^{(\text{des})}$  と衝突が発生しないように生成される必要がある。ここでは、生成された参照姿勢変数ベクトル  $\eta^{(\text{des})}$  に応じた静的障害物領域集合  $\mathcal{O}$  の生成手法について述べる。この手法で生成される障害物は船舶長さよりも長い線分のみで構成される。具体的に、以下の手順により障害物は生成される。

- まず、 $x_0y_0$  平面に格子点を設定する。ただし、格子点の間隔は  $2L_{\text{ship}}$ 、参照姿勢変数ベクトル  $\eta^{(\text{des})}$  の初期位置座標が格子点の一つとなるように設定される。生成される擬似障害物と目的の軌道との距離は格子点の間隔によって制御することができるが、環境が厳しくなりすぎないように格子点の間隔を  $2L_{\text{ship}}$  とする。
- 次に、障害物が存在しない領域を参照姿勢変数ベクトル  $\eta^{(\text{des})}$  に基づいて格子点で囲まれた領域の中から選択する。ここでは、船体の擬似的な形状が長軸半径が  $0.75L_{\text{ship}}$ 、短軸半径が  $B_{\text{ship}}$  の楕円で表現されると仮定し、参照姿勢変数ベクトル  $\eta^{(\text{des})}$  にしたがって船体が運動した場合に、船体が通過する領域を障害物が存在しない領域とする。また、初期追従誤差に起因する衝突を避けるため、参照姿勢変数

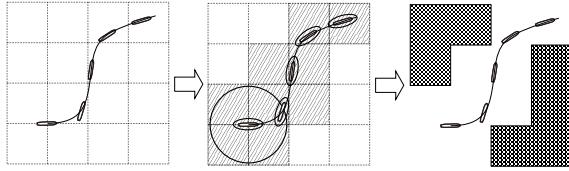


Fig. 5.4: The generation method of static pseudo-obstacles.

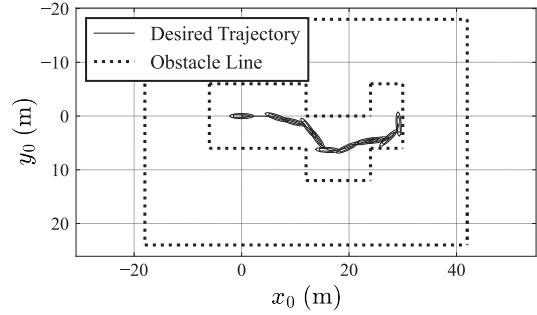


Fig. 5.5: An example of a desired trajectory with the static pseudo-obstacles used in training.

ベクトル  $\eta^{(\text{des})}$  の初期位置座標を中心とする半径  $1.9L_{\text{ship}}$  の円形領域においても障害物が存在しない領域とする。

この生成手順の概略図を Fig. 5.4に示す。また、この手法によって得られた静的障害物領域集合  $O$  の例を Fig. 5.5に示す。

### 5.2.3.3 報酬関数と終了条件

次に、報酬関数  $r$  における静的障害物に関するペナルティとエピソードの終了条件について説明する。軌道追従制御のための制御方策関数  $\mu$  に静的障害物を考慮させる目的は、追従誤差の発生を避けられない場合であつたとしても障害物との衝突を引き起こす可能性のある追従誤差を優先的に回避する制御方策関数  $\mu$  を獲得するためである。よって、障害物との衝突につながる可能性のある追従誤差に応じた報酬のペナルティを定義する。

#### (1) 衝突につながる追従誤差ペナルティ

まず、Fig. 5.6に示すような仮想の障害物線を導入する。この線は  $\mathbf{o}^{(\text{near})}$  を通り、参照姿勢変数ベクトル  $\eta^{(\text{des})}$  で表される船体と並行な直線である。また、この仮想の障害物線に直行する  $x_0y_0$  平面の法線ベクトルを  $\mathbf{n} \in \mathbb{R}^2$  と表記すると、この法線ベクトルは

$$\begin{bmatrix} \mathbf{n}(t) \\ 0 \end{bmatrix} = \begin{bmatrix} \mathbf{p}^{(\text{bow})}(\eta^{(\text{des})}(t)) - \mathbf{p}^{(\text{stn})}(\eta^{(\text{des})}(t)) \\ 0 \end{bmatrix} \times \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}, \quad (5.2.20)$$

と表される。ここで、 $\times$  は外積を表していることに注意する。このとき、仮想の障害物線と参照姿勢変数ベクトル  $\eta^{(\text{des})}$  の距離は

$$l^{(\text{des})}(t) = \left| \frac{\mathbf{n}(t) \cdot \left( \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \eta^{(\text{des})}(t) - \mathbf{o}^{(\text{near})}(t) \right)}{\|\mathbf{n}(t)\|} \right| \quad (5.2.21)$$

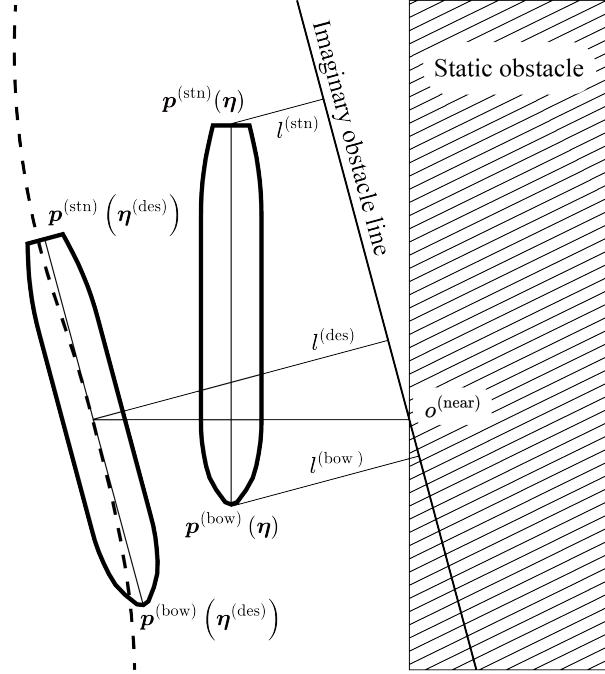


Fig. 5.6: Length between ship and static obstacle.

と表される。また、仮想の障害物線と船首および船尾の距離は

$$\begin{cases} l^{(\text{bow})}(t) = \left| \frac{\mathbf{n}(t) \cdot (\mathbf{p}^{(\text{bow})}(t) - \mathbf{o}^{(\text{near})}(t))}{\|\mathbf{n}(t)\|} \right| \\ l^{(\text{stn})}(t) = \left| \frac{\mathbf{n}(t) \cdot (\mathbf{p}^{(\text{stn})}(t) - \mathbf{o}^{(\text{near})}(t))}{\|\mathbf{n}(t)\|} \right| \end{cases} \quad (5.2.22)$$

と表される。よって、障害物との衝突につながる可能性のある追従誤差の尺度は

$$\begin{cases} c^{(\text{bow})}(t) = \max \left\{ 0, \frac{l^{(\text{des})}(t) - l^{(\text{bow})}(t)}{l^{(\text{des})}(t)} \right\} \\ c^{(\text{stn})}(t) = \max \left\{ 0, \frac{l^{(\text{des})}(t) - l^{(\text{stn})}(t)}{l^{(\text{des})}(t)} \right\} \end{cases} \quad (5.2.23)$$

と定義される。本研究ではこれらの尺度が衝突につながる可能性のある追従誤差ペナルティとして使用される。

## (2) 終了条件

エピソードを終了させる追従誤差の閾値について述べる。エピソードは障害物との衝突が発生するか、もしくは、 $c^{(\text{bow})}$  および  $c^{(\text{stn})}$  が閾値を超えた場合に終了させる。ただし、閾値はエピソード時間に応じて減少するように設計される。 $c^{(\text{bow})}$  および  $c^{(\text{stn})}$  の閾値を  $c^{(\text{tol})}$  と表記すると、エピソード時間  $t$  を用いて閾値  $c^{(\text{tol})}$  は

$$c^{(\text{tol})}(t) = \left( c_0^{(\text{tol})} - c_{\infty}^{(\text{tol})} \right) \exp(-\lambda_c t) + c_{\infty}^{(\text{tol})}, \quad (5.2.24)$$

と定義される。ここで、 $c_0^{(\text{tol})}$ ,  $c_\infty^{(\text{tol})}$ ,  $\lambda_c$  はそれぞれ、上限値の初期値、終端値、時間減少率を表す定数である。従って、 $c^{(\text{bow})}$  および  $c^{(\text{stn})}$  が閾値  $c^{(\text{tol})}$  より大きくなると、エピソードが終了され、それ以降の時刻では報酬が与えられない。

### 5.3 未知の参照軌道および静的障害物への適用

5.2 節では、強化学習を用いた軌道追従制御のための制御方策の獲得手法について述べた。獲得された最適制御方策関数  $\mu^*$  は順伝播型ニューラルネットワークによって表現されている。そのため、訓練で経験しなかった未知の状態変数ベクトル  $s$  に遭遇すると最適制御方策関数  $\mu^*$  の制御性能が低下する可能性がある。本節では、未知の参照軌道および静的障害物が与えられた場合であって、訓練で経験したものに近い状態変数ベクトル  $s$  を最適制御方策関数  $\mu^*$  へ与えるための工夫について詳述する。

#### 5.3.1 参照状態の選択方法

最適制御方策関数  $\mu^*$  は参照姿勢変数ベクトル  $\eta^{(\text{des})}$  を追従するように訓練されているが、必ずしも追従誤差を吸収可能であるとは限らない。例えば、外乱による力が船舶が生み出せる推力より大きい場合、訓練で経験しないような大きな追従誤差が発生してしまう可能性がある。このような状況では、最適制御方策関数  $\mu^*$  には未知の状態変数ベクトル  $s$  が入力され、最適制御方策関数  $\mu^*$  の制御性能が低下する可能性がある。したがって、本研究では、制御性能低下の可能性を軽減するため、参照姿勢変数ベクトル  $\eta^{(\text{des})}$  は時刻  $t$  によって決定せず、船体の姿勢変数ベクトル  $\eta$  に最も近い状態を時刻  $t$  の参照姿勢変数ベクトル  $\eta^{(\text{des})}$  を常に選択する。

ある離散時間ステップ  $t_k$  における船体の姿勢変数ベクトル  $\eta$  に最も近い参照姿勢変数ベクトル  $\eta^{(\text{des})}$  の時刻を  $\tau_k$  と表記すると、 $\tau_k$  は

$$\tau_k = \underset{\tau_{k-1} < \tau < \tau_{k-1} + T^{(\text{des})}}{\operatorname{argmin}} \left\| \eta^{(\text{des})}(\tau) - \eta(t_k) \right\|_W^2 \quad (5.3.1)$$

と定義される。ただし、 $T^{(\text{des})} = 1.0$  (s) で、 $W = \operatorname{diag}(1, 1, 0)$  とし、回頭角の誤差は無視する。このとき、Eq. (5.2.5) は

$$\mathbf{e}_T^{(\text{des})}(t_k) = \mathbf{R}^{-1}(\eta(t_k)) \left( \eta^{(\text{des})}(\tau_k + T) - \eta(t_k) \right) \quad (5.3.2)$$

と再定義される。従って、最適制御方策関数  $\mu^*$  の入力変数は Eq. (5.3.2) で求められた  $T$  時間先の追従姿勢誤差ベクトル  $\mathbf{e}_T^{(\text{des})}$  によって構成される。

#### 5.3.2 擬似障害物の生成

最適制御方策関数  $\mu^*$  は 5.2.3.2 節で述べた手法で生成された障害物に対して訓練されている。よって、訓練では  $T$  時間先の参照姿勢変数ベクトル  $\eta^{(\text{des})}$  から最も近い障害物点との位置誤差ベクトル  $\mathbf{e}_T^{(\text{obs})}$  のノルムは大きくとも  $2L_{\text{ship}}$  程度の値であるが、与えられる静的障害物領域集合  $O$  に依るがより大きな値が入力される可能性がある。そのため、本研究では、制御性能低下の可能性を軽減するため、与えられる静的障害物領域集合  $O$  に加えて 5.2.3.2 節で述べた手法で生成された擬似障害物を考慮した。ただし、5.2.3.2 節で述べた手法を直接使用すると、最適制御方策関数  $\mu^*$  の制御性能に影響を与える擬似障害物を生成する可能性がある。そのため、障害物が存在しない領域を選択するために用いられる船体の擬似的な形状を半径が  $1.9L_{\text{ship}}$  の円に変更する。また、ここで述べた手法により生成された擬似障害物の例を Fig. 5.7 に示す。

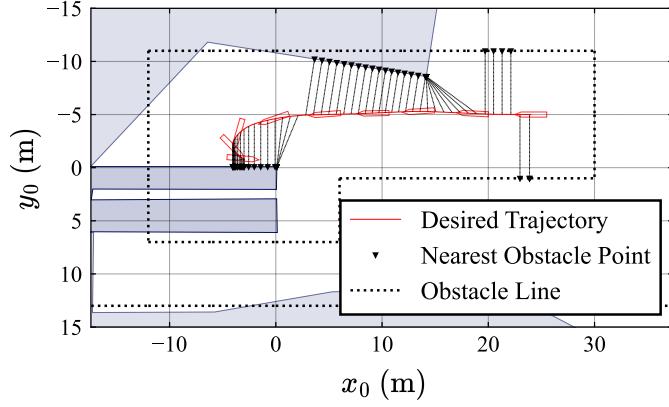


Fig. 5.7: A sample of the generated pseudo-obstacles in the target harbor.

## 5.4 MMG モデルを用いた検証実験

本節では、5.2 節で述べた軌道追従制御方策の訓練手法の有効性を示すため、MMG モデルで構成される操縦シミュレーション環境を用いて軌道追従制御方策を獲得し、シミュレーションおよび模型試験における与えられた着桟操船軌道の追従実験結果を示す。2.4 節で述べた供試船 2 を対象船舶とした。ただし、プロペラ回転数が  $n_P = 10.0$  (rps) で固定された。使用された MMG モデルはの詳細は2.3.2 節を参照されたい。ここで示される内容は、自著論文 3 [104] で示された内容と同様である。以降、5.4.1 節で軌道追従制御方策の訓練結果について、5.4.2 節で与えられる着桟操船軌道について、5.4.3 節で与えられた着桟操船軌道の追従実験結果について示す。

### 5.4.1 追従制御方策の訓練結果

本実験では、静的障害物領域集合  $O$  を考慮しない制御方策関数  $\mu$  と考慮した制御方策関数  $\mu$  の訓練を実施した。ここでは、前者を Ctrl-w/o-OBST、後者を Ctrl-w/-OBST と表記する。静的障害物領域集合  $O$  を考慮するかどうかによって、それぞれ使用された状態変数ベクトル  $s$  および報酬関数  $r$  が異なる。Ctrl-w/o-OBST では、状態変数ベクトル  $s$  は以下のように定義された。

$$s \equiv \left( \left( s^{(\text{des})} \right)^T, v^T, u^T \right)^T, \quad (5.4.1)$$

また、報酬関数  $r$  は以下のように定義された。

$$r(s_{k+1}, s_k, a_k) \equiv 2 - \frac{e^{(\text{bow})}(t_k) + e^{(\text{stn})}(t_k)}{e^{(\text{tol})}(t_k)} - \lambda \sum_{i=1}^{N_u} \left( \frac{a_i - u_{c,i}}{u_{\text{std},i}} \right)^2, \quad (5.4.2)$$

一方で Ctrl-w/-OBST では、状態変数ベクトル  $s$  は以下のように定義された。

$$s \equiv \left( \left( s^{(\text{des})} \right)^T, \left( s^{(\text{obs})} \right)^T, v^T, u^T \right)^T, \quad (5.4.3)$$

Table 5.1: Hyperparameters for Ctrl-w/o-OBST and Ctrl-w/-OBST.

$\Delta t = 0.1$ (s)	$\sigma_{x_0}^{(\text{obs})} = 0.03$ (m)	$e_0^{(\text{tol})} = 2L_{\text{ship}}$ (m)	$u_{c,1} = -75$ (deg.)
$t_{k+1} - t_k = 5.0$ (s)	$\sigma_{y_0}^{(\text{obs})} = 0.03$ (m)	$e_{\infty}^{(\text{tol})} = B_{\text{ship}}/2$ (m)	$u_{c,2} = 75$ (deg.)
$\gamma = 0.99$	$\sigma_{\psi}^{(\text{obs})} = 0.2$ (deg.)	$\lambda_e = (\log 2)/50$	$u_{c,3} = 0.0$ (deg.)
$\lambda = 1/300$	$\sigma_u^{(\text{obs})} = 0.02$ (m/s)	$c_0^{(\text{tol})} = 1.0$	$u_{\text{std},1} = 110$ (deg.)
$T_1 = 5.0$ (s)	$\sigma_{v_m}^{(\text{obs})} = 0.02$ (m/s)	$c_{\infty}^{(\text{tol})} = 0.5$	$u_{\text{std},2} = 110$ (deg.)
$T_2 = 10.0$ (s)	$\sigma_r^{(\text{obs})} = 0.2$ (deg./s)	$\lambda_c = (\log 2)/50$	$u_{\text{std},3} = 30.0$ (deg.)
$T_3 = 20.0$ (s)	$\sigma_u^{(\text{sys})} = 1.0^{-4}$ (m/s)		
$T_4 = 40.0$ (s)	$\sigma_{v_m}^{(\text{sys})} = 1.0^{-4}$ (m/s)		
	$\sigma_r^{(\text{sys})} = 1.0^{-3}$ (deg./s)		

また、報酬関数  $r$  は以下のように定義された。

$$r(s_{k+1}, s_k, \mathbf{a}_k) \equiv 4 - \frac{e^{(\text{bow})}(t_k) + e^{(\text{stn})}(t_k)}{e^{(\text{tol})}(t_k)} - \frac{c^{(\text{bow})}(t_k) + c^{(\text{stn})}(t_k)}{c^{(\text{tol})}(t_k)} - \lambda \sum_{i=1}^{N_u} \left( \frac{a_i - u_{c,i}}{u_{\text{std},i}} \right)^2, \quad (5.4.4)$$

また、訓練で用いられたハイパーパラメータを Table 5.1に、操縦シミュレーション環境の初期状態変数の決定方法を Table 5.2に示す。ここで、 $e_0^{(\text{tol})}$  の値は、初期追従誤差のほぼ 2 倍に設定されている。また、 $\lambda_e$  と  $\lambda_c$  の値は、50 秒後に許容範囲が半分になるように設定した。さらに、TD3 のハイパーパラメータは、バッヂサイズは 512、他は文献 [102] の Table 3 の値が使用され、アクターネットワークとクリティックネットワークの構造を Table 5.3に示す。訓練過程に含まれるランダム性が訓練結果に与える影響を確認するため、訓練は Ctrl-w/-OBST と Ctrl-w/o-OBST のそれぞれ 5 回ずつ実施した。訓練時間はシミュレーション時間で  $3.0 \times 10^7$  秒である。

本実験では、学習中に取得した全ての NN のパラメータを保存し、それぞれのパラメータについて評価を行った。評価では、学習エピソードで使用したのと同じ環境で 20 エピソードをシミュレーションし、平均累積報酬を算出した。5 つの学習に対する平均累積報酬を Fig. 5.8 に示す。この結果から、最適制御方策関数  $\mu^*$  は訓練に含まれるランダム性に大きく依存しないことがわかる。そこで、5 つの訓練の中で最も高い累積報酬を得たパラメータが、最も適切な方法として選択された。Ctrl-w/-OBST と Ctrl-w/o-OBST の累積報酬は、報酬関数  $r$  が異なるため、直接比較に意味はないことに注意する。

#### 5.4.2 着桟操船の軌道計画

軌道追従制御方策に与える着桟操船軌道について述べる。着桟操船の参考軌道は宮内ら [78] によって提案された着桟操船の軌道計画手法に従って生成された。このアルゴリズムでは、着桟操船の軌道計画は時間最小化問題としてモデル化され、空間制約と終端条件は目的関数に組み込まれている。そして、この数理最適化問題は CMA-ES [129] を用いて解かれる。

本実験では、4 つの異なる終端条件と 11 の異なる初期条件を持つ 44 の軌道を準備した。使用した初期条件および終端条件は Table 5.4に示す。終端条件の目標姿勢は、バースから  $1.5B_{\text{ship}}$  だけ離れた地点で定義され、異なるヘディング角を持つ。初期条件の船舶姿勢は、目標姿勢から約  $8L_{\text{ship}}$  離れた地点に設定された。初期条件の船舶速度は、目標地点に到達する前に十分に減速できる速度として決定された。終端条件の許容値は、文献 [78] に基づいて決定された。軌道計画では、アクチュエータ指令値の限界値を、制御力の余裕を十分に持

Table 5.2: Initial state of maneuvering simulation environments.

Item	Definition
$x_0^{(\text{des})}$ (m)	is set to 0.0
$y_0^{(\text{des})}$ (m)	is set to 0.0
$\psi^{(\text{des})}$ (deg.)	is set to 0.0
$u^{(\text{des})}$ (m/s)	is given by uniform distribution whose interval is $[-0.072, 0.437]$
$v_m^{(\text{des})}$ (m/s)	is given by uniform distribution whose interval is $[-0.07, 0.07]$
$r^{(\text{des})}$ (deg./s)	is given by uniform distribution whose interval is $[-0.1, 0.1]$
$x_0$ (m)	is given by uniform distribution whose interval is $[x_0^{(\text{des})} - L_{\text{ship}}, x_0^{(\text{des})} + L_{\text{ship}}]$
$y_0$ (m)	is given by uniform distribution whose interval is $[y_0^{(\text{des})} - L_{\text{ship}}, y_0^{(\text{des})} + L_{\text{ship}}]$
$\psi$ (deg.)	is given by uniform distribution whose interval is $[\psi^{(\text{des})} - 10, \psi^{(\text{des})} + 10]$
$u$ (m/s)	is given by uniform distribution whose interval is $[u^{(\text{des})} - 0.036, u^{(\text{des})} + 0.036]$
$v_m$ (m/s)	is given by uniform distribution whose interval is $[v_m^{(\text{des})} - 0.007, v_m^{(\text{des})} + 0.007]$
$r$ (deg./s)	is given by uniform distribution whose interval is $[r^{(\text{des})} - 0.1, r^{(\text{des})} + 0.1]$
$\bar{U}_T$ (m)	is given by Weibull distribution whose shape and scale parameter are 2.0 and 1.0
$\bar{\xi}_T$ (deg.)	is given by uniform distribution in $\mathbb{S}$

Table 5.3: Used Layers for networks. Values in brackets express the inputs into Ctrl-w/o-OBST.

(a) The actor network (policy function).

	$W_l$	$b_l$	$g$
$\mathbf{h}_1^{(\text{fcl})}$	$\mathbb{R}^{256 \times 32}$ ( $\mathbb{R}^{256 \times 22}$ )	$\mathbb{R}^{256}$	tanh
$\mathbf{h}_2^{(\text{fcl})}$	$\mathbb{R}^{256 \times 256}$	$\mathbb{R}^{256}$	tanh
$\mathbf{h}_3^{(\text{fcl})}$	$\mathbb{R}^{256 \times 256}$	$\mathbb{R}^{256}$	tanh
$\mathbf{h}_4^{(\text{fcl})}$	$\mathbb{R}^{3 \times 256}$	$\mathbb{R}^3$	Sigmoid

(b) The critic network (Q function).

	$W_l$	$b_l$	$g$
$\mathbf{h}_1^{(\text{fcl})}$	$\mathbb{R}^{256 \times 35}$ ( $\mathbb{R}^{256 \times 25}$ )	$\mathbb{R}^{256}$	tanh
$\mathbf{h}_2^{(\text{fcl})}$	$\mathbb{R}^{256 \times 256}$	$\mathbb{R}^{256}$	tanh
$\mathbf{h}_3^{(\text{fcl})}$	$\mathbb{R}^{256 \times 256}$	$\mathbb{R}^{256}$	tanh
$\mathbf{h}_4^{(\text{fcl})}$	$\mathbb{R}^{1 \times 256}$	$\mathbb{R}^1$	None

った軌道を生成するために、Table 5.5で示される限界値が用いられた。この考え方は、小瀬ら [61] によって提案されている。得られた軌道を Fig. 5.9に示す。

### 5.4.3 着桟操船軌道の追従結果

訓練で得られた最適制御方策関数  $\mu^*$  を用いた着桟操船起動の追従制御の結果を示す。本実験では、MMG モデルベースの操縦シミュレーション環境と物理環境での検証を実施した。ただし、検証における意思決定間隔は  $t_{k+1} - t_k = 1.0$  (s) に変更された。以降、操縦シミュレーション環境の結果を5.4.3.1 節で、物理環境の結果を5.4.3.2 節で述べる。

#### 5.4.3.1 操縦シミュレーション環境における検証

操縦シミュレーション環境、つまり、訓練時と同じ環境における Ctrl-w/-OBST および Ctrl-w/o-OBST の評価結果を示す。操縦シミュレーションは衝突が発生したとき、またはシミュレーションの経過時間が 250 秒

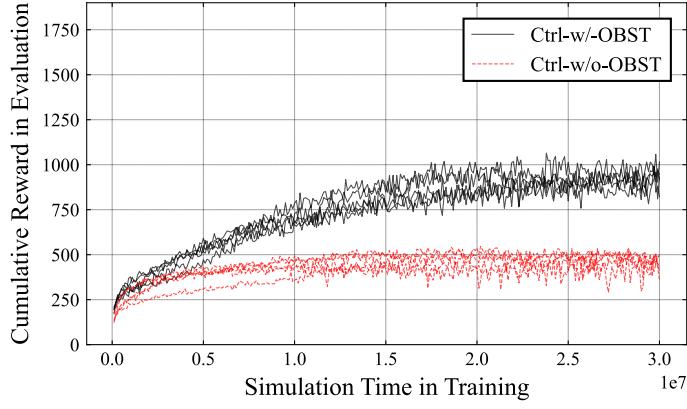


Fig. 5.8: Average cumulative reward of the 20 episodes used for the evaluation.

Table 5.4: Initial and terminal conditions, and tolerances used in trajectory planning.

Symbol	Value
$\mathbf{x}_{\text{init},1}$	$(8L_{\text{ship}}, -5.0, \pi, 0.291, 0.0, 0.0)^T$
$\mathbf{x}_{\text{init},2}$	$(8L_{\text{ship}}, -4.0, \pi, 0.291, 0.0, 0.0)^T$
$\vdots$	$\vdots$
$\mathbf{x}_{\text{init},10}$	$(8L_{\text{ship}}, 4.0, \pi, 0.291, 0.0, 0.0)^T$
$\mathbf{x}_{\text{init},11}$	$(8L_{\text{ship}}, 5.0, \pi, 0.291, 0.0, 0.0)^T$
$\mathbf{x}_{\text{term},1}$	$(-3.00, -1.5B_{\text{ship}}, \pi, 0.0, 0.0, 0.0)$
$\mathbf{x}_{\text{term},2}$	$(-3.00, -1.5B_{\text{ship}}, 0.0, 0.0, 0.0, 0.0)$
$\mathbf{x}_{\text{term},3}$	$(1.5B_{\text{ship}}, 3.0, 3\pi/2, 0.0, 0.0, 0.0)$
$\mathbf{x}_{\text{term},4}$	$(1.5B_{\text{ship}}, 3.0, \pi/2, 0.0, 0.0, 0.0)$
$\mathbf{x}_{\text{tol}}$	$(0.02, 0.02, 0.0175, 0.0141, 0.0141, 0.0094)$

Table 5.5: Limitations of the actuator state variables used in trajectory planning.

Symbols	Range
$\delta_P$ (deg.)	$[-105, -45]$
$\delta_S$ (deg.)	$[45, 105]$
$n_{BT}$ (rps)	$[-15, 15]$

に達したときに終了した。ただし、衝突検出において、船体の形状は長軸半径が  $0.75L_{\text{ship}}$ 、短軸半径が  $B_{\text{ship}}$  の橢円であると仮定された。

### (1) 無風状態 $U_T = 0.0$ (m/s) における追従結果

まず、無風状態  $U_T = 0.0$  (m/s) における追従結果を Figs. 5.10–5.13に示す。ここで、 $y_0$  軸方向の初期誤差は 3.0 (m) であり、初期条件が  $\mathbf{x}_{\text{init},1}$  の着棧操船軌道が与えられている。

Fig. 5.10を見ると、終端条件が  $\mathbf{x}_{\text{term},1}$  の着棧操船軌道に対して、Ctrl-w/-OBST および Ctrl-w/o/OBST のいずれも初期誤差を吸収し、参照姿勢変数ベクトル  $\boldsymbol{\eta}^{(\text{des})}$  に沿った操船を実現していることがわかる。しかし、Ctrl-w/o/OBST を用いたシミュレーションは衝突により終了している。衝突は接岸に最も近い目標姿勢の付近で発生し、衝突直前に船首方位角の誤差が増加した。船首スラスターは  $y_0$  軸の誤差を低減するために右

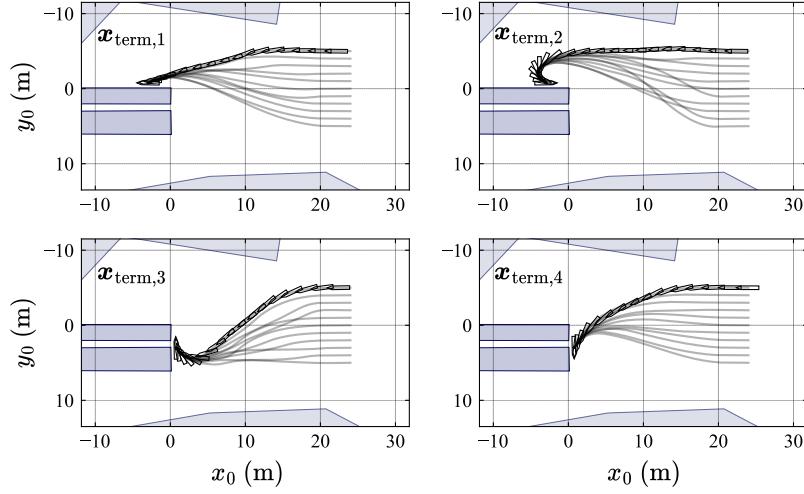


Fig. 5.9: Desired trajectories generated by the trajectory planning procedure.

舷方向の力を発生させ、舵角は船速を低減するために後進力を発生させるように選択された。特に、舵角の上限と下限 ( $\delta_P = -105$  (deg.),  $\delta_S = 105$  (deg.)) が継続して選択され、 $y_0$  軸方向の力を制御しなかった。その結果、バウスラスタによって発生したモーメントは舵によって相殺されず、方位角の誤差の原因となった。一方、Ctrl-w/-OBST では衝突することなくシミュレーションが終了し、方位角の誤差も小さく抑えられている。

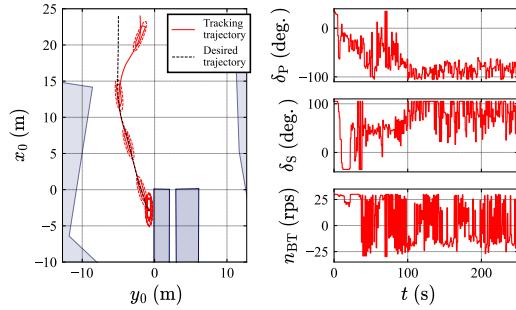
Fig. 5.12から、終端条件が  $x_{term,1}$  の着桟操船軌道に対して、Ctrl-w/o-OBST を用いたシミュレーションは衝突により終了した。この場合、コントローラは動揺とヨー速度をゼロに近づけたが、速度がゼロに達する前に衝突が発生した。つまり、 $x_0$  と  $\psi$  の小さなオーバーシュートが発生し、衝突を引き起こした。したがって、接岸操船では、追従制御器の性能が高くても、わずかな追従誤差により衝突が発生する。また、Figs. 5.11と5.13を見ると、いずれの制御器が接岸軌道を追従でき、衝突することなくシミュレーションが終了したことがわかる。

## (2) 衝突確率の比較結果

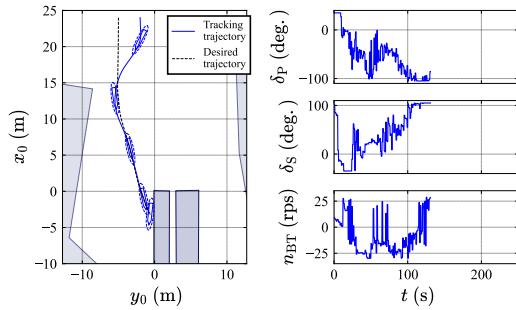
衝突の発生は、与えられる参考姿勢変数ベクトル  $\eta^{(des)}$ 、真風速風向ベクトル  $w_T$ 、初期追従誤差によって異なる可能性がある。これらの変化による影響を示すため、Ctrl-w/-OBST と Ctrl-w/o-OBST の着桟制御における衝突確率を比較し、Ctrl-w/-OBST の有効性を示す。ここで、衝突確率は、シミュレーションの終了前に静的障害物と衝突する確率として定義される。

衝突確率は、着桟操船軌道の終端条件と平均風速を固定し、初期追従誤差と平均風向を変化させて計算した。具体的には、100 回の着桟操船軌道の追従制御の試行を行い、初期追従誤差と平均風向を変化させ、着桟操船軌道の 11 種類の初期条件を用いて衝突確率を算出した。初期追従誤差および平均風向は Table 5.2 に示された方法によって確率的に決定され、観測ノイズおよびシステムノイズは Table 5.1 に示された通りに決定される。サンプルサイズが十分でない可能性があるため、衝突確率は 5 回計算し、その平均値と標準偏差を求めた。衝突確率の平均値と標準偏差は Table 5.6 に示す。ここでは、風圧や船舶が停止状態で生成可能な推力の上限を考慮し、平均風速が 1.5 m/s 以下の場合で衝突確率を計算した。

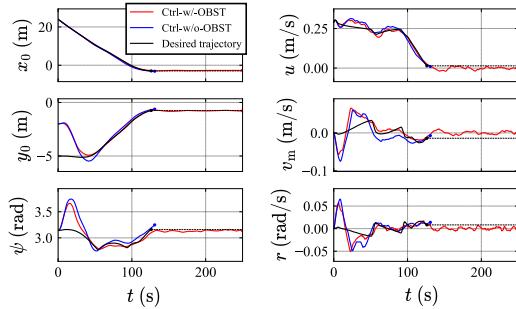
Ctrl-w/o-OBST を使用した場合、終端条件が  $\eta_{term,1}$  または  $\eta_{term,3}$  の着桟操船軌道で衝突確率が比較的高いことが確認された。小さな追従誤差によって衝突が発生し、平均風速が低い場合でも衝突確率が高くなるこ



(a) Trajectories and actions using Ctrl-w/-OBST



(b) Trajectories and actions using Ctrl-w/o-OBST

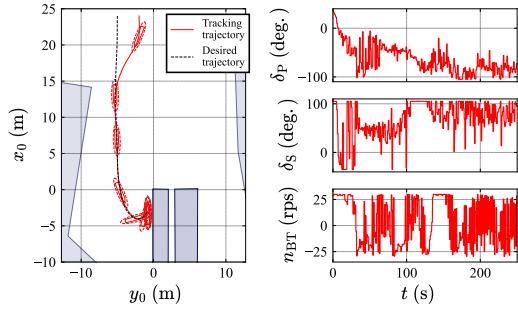


(c) The time series of the pose and velocity vector

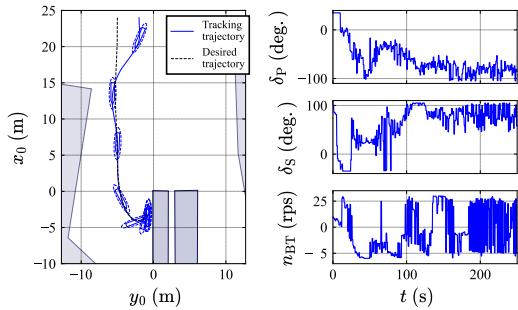
Fig. 5.10: Tracking results of the berthing trajectory in no wind condition. The terminal condition of the berthing trajectory is  $\eta_{\text{term},1}$ .

とが Figs. 5.10b と 5.12b に示されている。

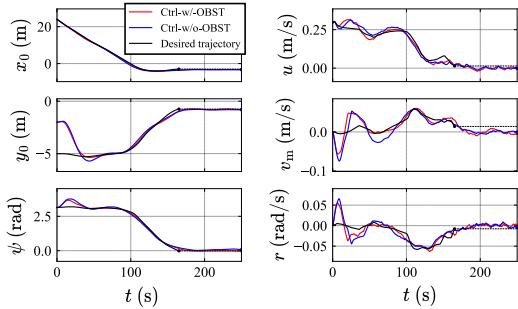
一方で、Ctrl-w/-OBST を使用した場合、終端条件が  $\eta_{\text{term},2}$  の着棧操船軌道で衝突確率が比較的高いことが確認された。この原因の一つとして、旋回時に船首側の衝突検知領域が障害物に接触することが挙げられる (Fig. 5.14)。これらの着棧操船軌道は、5.2.3.2 節で述べた衝突検知領域を考慮して生成されているため、旋回時のわずかな追従誤差が衝突を引き起こす。このため、衝突検知機能の長軸を  $0.5L$  に変更し、衝突確率を再計算した。その結果についても Table 5.6 に示されており、この条件下では他の終端条件と同様の衝突確率が



(a) Trajectories and actions using Ctrl-w/-OBST

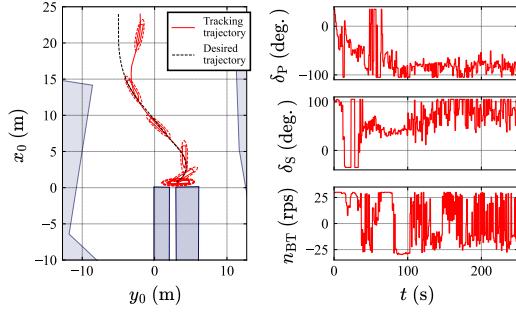


(b) Trajectories and actions using Ctrl-w/o-OBST

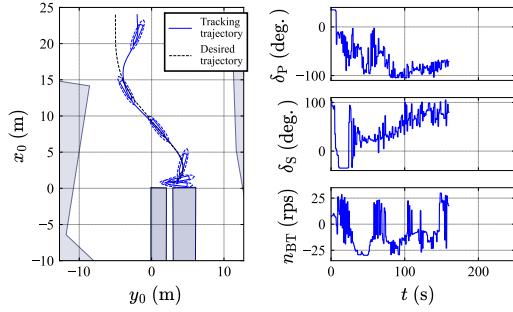


(c) The time series of the pose and velocity vector

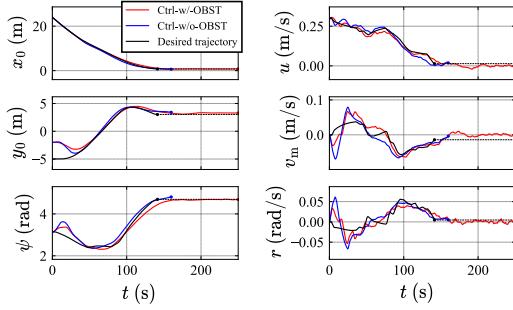
Fig. 5.11: Tracking results of the berthing trajectory in no wind condition. The terminal condition of the berthing trajectory is  $\eta_{\text{term},2}$ .



(a) Trajectories and actions using Ctrl-w/-OBST



(b) Trajectories and actions using Ctrl-w/o-OBST

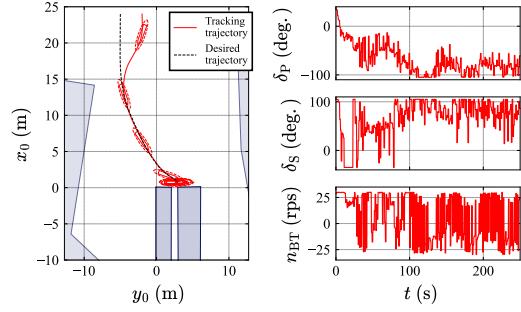


(c) The time series of the pose and velocity vector

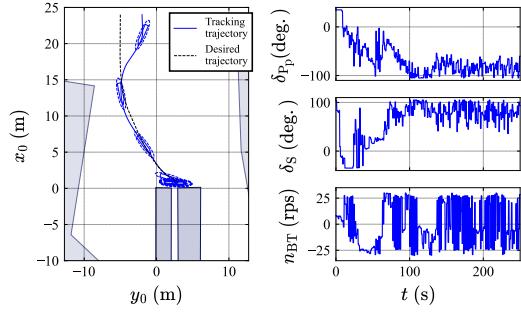
Fig. 5.12: Tracking results of the berthing trajectory in no wind condition. The terminal condition of the berthing trajectory is  $\eta_{\text{term},3}$ .

得られた。

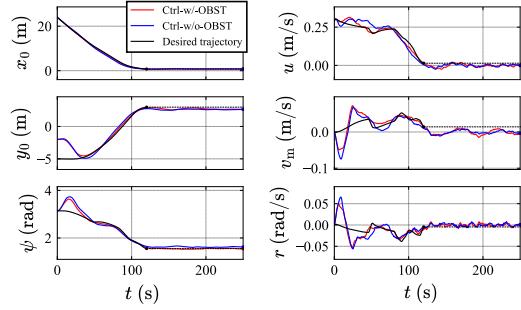
Ctrl-w/-OBST と Ctrl-w/o-OBST の衝突確率を比較すると、終端条件が  $\eta_{\text{term},2}$  の場合を除いて、Ctrl-w/-OBST の方が衝突確率が低いことが分かる。しかし、 $\eta_{\text{term},2}$  の終端条件の場合でも、Ctrl-w/-OBST の衝突確率は他の終端条件と同程度である。したがって、報酬関数で障害物との衝突につながる追従誤差に対してペナルティを与えることで、着桟操船軌道の追従制御における衝突確率を低減できることが示された。



(a) Trajectories and actions using Ctrl-w/-OBST



(b) Trajectories and actions using Ctrl-w/o-OBST



(c) The time series of the pose and velocity vector

Fig. 5.13: Tracking results of the berthing trajectory in no wind condition. The terminal condition of the berthing trajectory is  $\eta_{\text{term},4}$ .

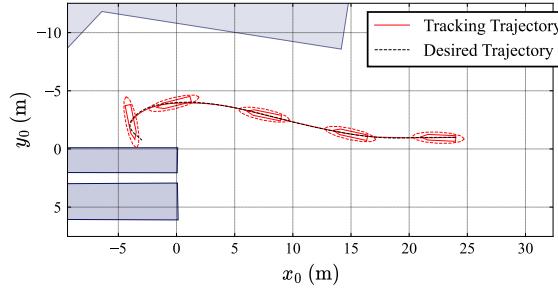


Fig. 5.14: A tracking result of the berthing trajectory using the Ctrl-w/-OBST. The terminal condition of the desired trajectory is  $\eta_{\text{term},2}$ , and mean wind speed  $\bar{U}_T$  is 0.0 (m/s).

Table 5.6: The mean values and standard deviations of the collision probabilities. Probabilities are expressed as percentages. Note that the second  $x_{\text{term},2}$  represents the collision probability with the semi-major axis of the collision detection ellipse changed to  $0.5L$ .

$\bar{U}_T$ (m/s)	Terminal condition of planned trajectory				
	$x_{\text{term},1}$	$x_{\text{term},2}$	$x_{\text{term},3}$	$x_{\text{term},4}$	$x_{\text{term},2}$
Ctrl-w/-OBST	0.0	$7.9 \pm 1.1$	$37.8 \pm 0.7$	$0.0 \pm 0.0$	$0.4 \pm 0.1$
	0.5	$6.7 \pm 2.0$	$39.0 \pm 1.8$	$0.0 \pm 0.0$	$0.3 \pm 0.2$
	1.0	$20.2 \pm 5.2$	$42.6 \pm 4.2$	$1.1 \pm 0.8$	$3.7 \pm 0.9$
	1.5	$45.2 \pm 7.3$	$48.7 \pm 4.7$	$38.4 \pm 4.1$	$37.5 \pm 2.8$
Ctrl-w/o-OBST	0.0	$94.3 \pm 0.6$	$1.0 \pm 0.2$	$95.4 \pm 0.9$	$3.2 \pm 0.4$
	0.5	$90.8 \pm 1.1$	$6.4 \pm 0.5$	$92.7 \pm 0.7$	$3.1 \pm 0.6$
	1.0	$72.6 \pm 3.9$	$30.0 \pm 3.4$	$70.1 \pm 3.6$	$6.4 \pm 1.4$
	1.5	$63.6 \pm 5.6$	$48.9 \pm 4.4$	$73.2 \pm 5.6$	$52.6 \pm 4.1$

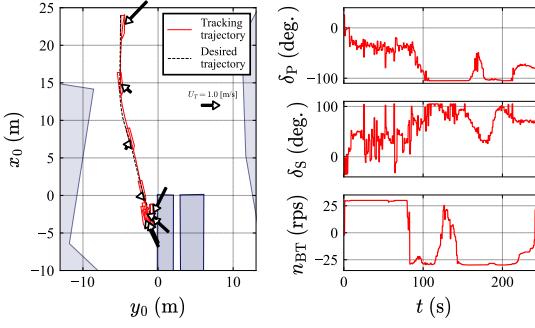
#### 5.4.3.2 模型試験による検証

次に、模型船を用いた物理環境における Ctrl-w/-OBST の評価結果を示す。実験は2.5 節で述べた実験池で実施され、2.4.1 節で述べた計測システムによって観測と制御が行われた。

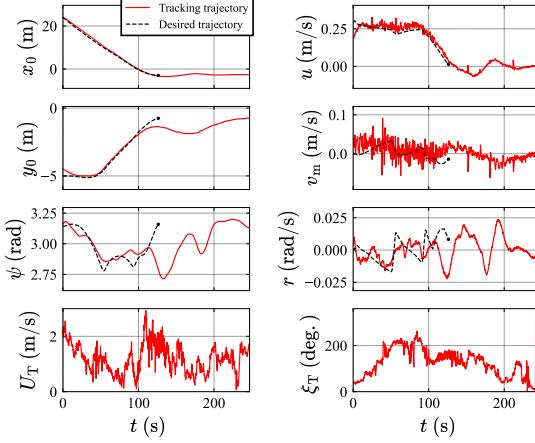
ここでは、終端条件が  $\eta_{\text{term},1}$  および  $\eta_{\text{term},2}$  の着棧軌道が使用され、それらに関連する追従結果は Fig. 5.15 および Fig. 5.16 に示されている。

Fig. 5.15の場合、模型船は  $t = 100$  (s) までの接近フェーズで着棧軌道を追従することができた。しかし、 $t = 100$  (s) 以降、模型船は軌道から逸脱したことが観察された。ただし、Ctrl-w/-OBST は VecTwin 舵による後進力とバウスラスタによる負の横力を生成し、模型船を所望の軌道に戻すことを試みていることがわかる。したがって、この追従誤差は風速の増加に起因するものと考えられる。

この場合、模型船は衝突せずに着棧地点に到達したが、風向が異なる場合には衝突の確率が増加する可能性がある。そのため、この風速では着棧操作を中止する決定が必要となる。今後、安全な着棧操作が可能な風速に関するさらなる研究が求められる。

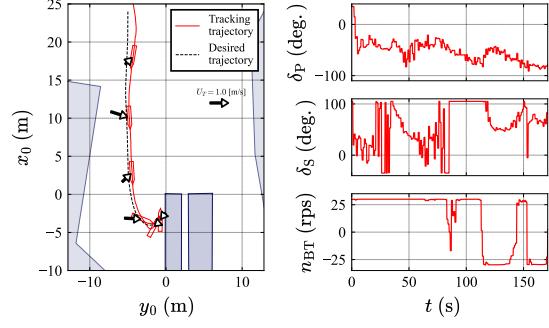


(a) Trajectories and the time series of control inputs

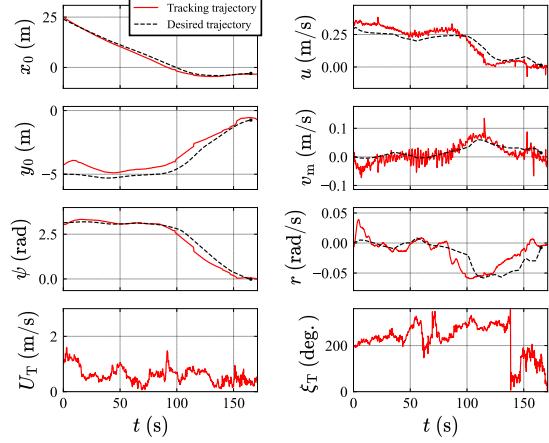


(b) The time series of the pose and velocity vector

Fig. 5.15: A result of model experiments tracking the berthing trajectory using the Ctrl-w/-OBST algorithm. The terminal condition of the desired trajectory was  $\eta_{\text{term},1}$ , and wind conditions were  $\bar{U}_T = 1.19$  (m/s),  $\bar{\xi}_T = 0.69\pi$  (rad).



(a) Trajectories and the time series of control inputs



(b) The time series of the vessel pose and velocity

Fig. 5.16: A result of model experiments tracking the berthing trajectory using the Ctrl-w/-OBST algorithm. The terminal condition of the desired trajectory was  $\eta_{\text{term},2}$ , and wind conditions were  $\bar{U}_T = 0.60$  (m/s),  $\bar{\xi}_T = -0.62\pi$  (rad).

一方、Fig. 5.16では、Ctrl-w/-OBST が旋回を含む着桟軌道を追従し、着桟地点付近で船速を許容速度  $v_{\text{tol}}$  に近い値まで減少させることができたことが示されている。この場合、風の条件は比較的穏やかであった。

## 5.5 操縦運動データを用いた実験

本節では、第 3 章で述べた ANN を用いた SI 手法を用いて操縦運動データから操縦モデルを推定し、得られた操縦モデルで構成される操縦シミュレーション環境を用いて軌道追従制御方策を獲得し、着桟操船軌道の追従実験および 4 コーナー DP 試験の結果を示す。本実験では、MMG モデルを用いた操縦シミュレーション環境を Target system (TS) としたことに注意する。ここで示される内容は、自著論文 3 [104] で示された内容と同様である。

Table 5.7: Informations for given dataset  $\mathcal{D}^{(\text{traj})}$ .

Item	Symbol	Value
Number of observed trajectories	$N$	24
The time step of a trajectory	$\Delta t$	1.0 (s)
Number of time steps in a trajectory	$K$	250
Total time of the dataset		6,000 (s)

本実験で与えられる操縦運動データは TS 環境における複数回の操縦シミュレーションで生成された。シミュレーションは停止状態から開始され、5.2.2.2 節で説明された舵角およびバウスラスタ回転数がランダムに選択される操船方法が用いられた。獲得した操縦運動データ  $\mathcal{D}^{(\text{traj})}$  に関するパラメータを Table 5.7 に示す。使用されたデータ集合  $\mathcal{D}$  の総時間は 6,000 (s) である。操縦モデルの予測精度が制御方策の性能に与える影響を確認するため、提案手法で得られた制御方策に加えて、TS 環境で直接訓練された軌道追従のための制御方策を獲得し、比較を行う。

以降、5.5.1 節では操縦運動データ集合  $\mathcal{D}^{(\text{traj})}$  を用いた操縦モデルの推定結果について、5.5.3 節で与えられた軌道追従実験の結果について示す。

### 5.5.1 操縦モデルの推定結果

操縦モデルの推定結果を示す。本実験では、4.2 節で用いられた FNN と訓練手法によって操縦モデルが推定された。つまり、Table 4.4 に示されたハイパーパラメータが使用され、Table 4.3 に示された全結合層のみで構成される多層パーセプトロンが用いられた。また、FNN の初期パラメータやミニバッチの順序のような訓練における不確実性による、最適モデルの変化の大きさを確認するため、10,000 Epochs の訓練を 5 回実施し、5 つの最適操縦モデルを獲得した。

対象システムと操縦モデルの操縦運動軌道を、ジグザグ操船および着桟操船を通して、比較する。ただし、着桟操船とは、フルケール版の対象船の着桟のための港湾内ナビゲーションデータ [126] を基にして、舵およびバウスラスタの指示値が決定された操船のことであり、対象システムを用いて着桟のための制御が行われているわけではないことに注意する。

ジグザグ操船のシミュレーション結果を Fig. 5.17 に示す。30°-30° のジグザグ操船では、最適モデルは  $u$  をやや大きく推定しており、その結果、軌道の推定誤差が時間経過とともに増大している。しかしながら、 $r$  と  $v_m$  の推定値は対象システムのものと高い精度で一致しており、短期的には軌道の推定誤差も十分に高い精度で推定できている。また、訓練ケース間で比較しても、特定のケースの推定精度が他に比べて大きく異なることもないことがわかる。

次に、着桟操船のシミュレーション結果を Fig. 5.18 に示す。ジグザグ操船に比べて着桟操船ではアクチュエータ状態が頻繁に変更されており、複雑な操縦運動が行われている。そのため、 $r$  と  $v_m$  の推定値は対象システムのものと高い精度で常に一致しているわけではなかったが、大きく乖離することはないことがわかる。また、着桟操船では、船速が約 0.5 (m/s) から約 0.0 (m/s) まで減少しているが、最適モデルはその船速の減少を適切に評価できている。したがって、低速運動を含む着桟操船でも最適モデルは十分な推定精度を持っていることがわかる。

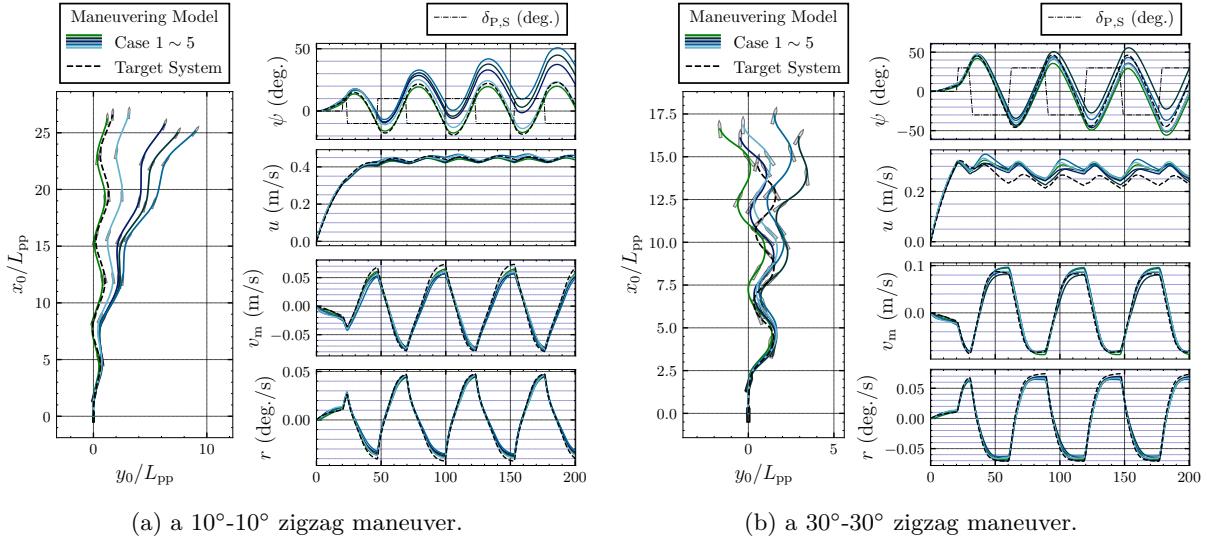


Fig. 5.17: The predicted trajectories of maneuvering simulation for zigzag maneuvers in calm wind condition  $U_T = 0.0$  (m/s). The solid colored lines represent the results using the identified maneuvering model in each case. The pentagons represent ship positions and headings.

### 5.5.2 追従制御方策の訓練結果

本実験では、操縦運動データ集合  $\mathcal{D}^{(\text{traj})}$  を用いて得られた操縦モデルで構成される操縦シミュレーション環境と TS 環境のそれぞれで軌道追従制御方策の訓練を実施した。ここでは、問題を簡単化するため、静的障害物領域集合  $\mathcal{O}$  は考慮されなかった。使用された状態変数ベクトル  $s$  および報酬関数  $r$  もまた5.4 節と異なり、状態変数ベクトル  $s$  は以下のように定義された。

$$s \equiv \left( \left( s^{(\text{des})} \right)^T, v^T, u^T, (w'_A)^T \right)^T, \quad (5.5.1)$$

また、報酬関数  $r$  は以下のように定義された。

$$r(s_{k+1}, s_k, a_k) \equiv 1 - \frac{(e^{(\text{bow})}(t_{k+1}) + e^{(\text{stn})}(t_{k+1}))}{2e^{(\text{tol})}(t_{k+1})} - (a_k - u(t_k))^T R (a_k - u(t_k)), \quad (5.5.2)$$

ここで、 $R = \text{diag}(R_P, R_S, R_{BT})$  はアクチュエータコストの重み係数である。訓練で用いられたハイパーパラメータを Table 5.8 に、操縦シミュレーション環境の初期状態変数の決定方法を Table 5.9 に示す。ここで、 $\mathcal{D}^{(\text{traj})}$  に含まれる状態を中心に訓練するため、生成される参照軌道の初期状態の決定方法が5.4.1 節から変更されていることに注意する。さらに、TD3 のハイパーパラメータは、バッチサイズは 256、他は文献 [102] の Table 3 の値が使用され、アクターネットワークとクリティックネットワークの構造は Table 5.10 に示される。訓練過程に含まれるランダム性が訓練結果に与える影響を確認するため、5 つの操縦モデルそれぞれに対して軌道追従制御方策の訓練を実施し、5 つの最適方策を獲得し、また、TS 環境からも 5 つの最適方策を訓練した。訓練時間はシミュレーション時間で  $1.5 \times 10^7$  秒である。

本実験では、学習中に取得した全ての NN のパラメータを保存し、それぞれのパラメータについて評価を行った。評価では、学習エピソードで使用したのと同じ環境で 10 エピソードをシミュレーションし、平均累積

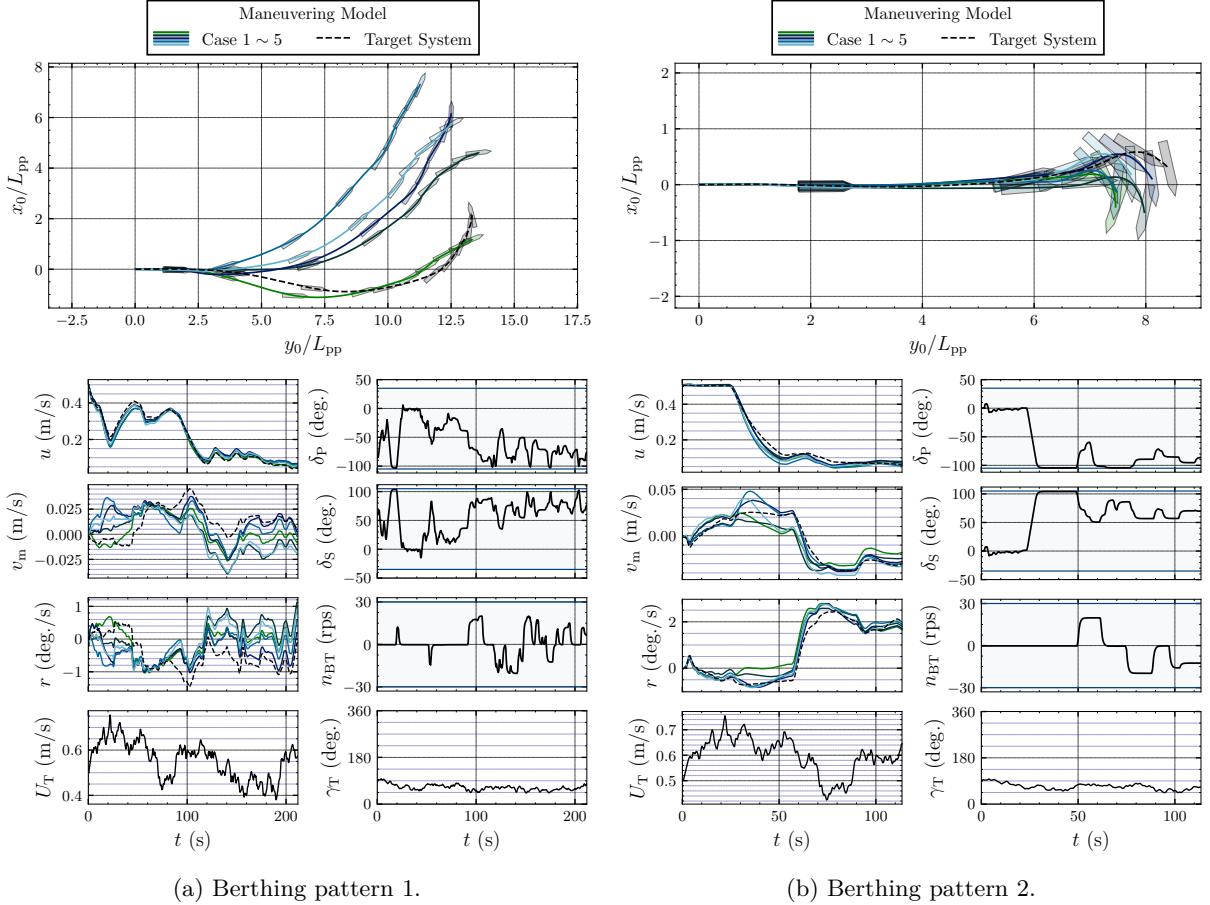


Fig. 5.18: The predicted trajectories and time histories of maneuvering simulation for a berthing maneuver under wind disturbance,  $\bar{U}_T = 0.5$  (m/s),  $\bar{\xi}_T = 90$  (deg.). The solid colored lines represent the results using the identified maneuvering model in each case. The pentagons represent ship positions and headings.

報酬を算出した。5つの学習に対する平均累積報酬を Fig. 5.19に示す。この結果から、最適制御方策関数  $\mu^*$  は訓練に含まれるランダム性に大きく依存しないことがわかる。

### 5.5.3 参照軌道の追従結果

訓練で得られた最適制御方策関数  $\mu^*$  を用いた着桟操船軌道の追従実験および4コーナーDP試験の結果を示す。本実験では、TS環境で検証を実施した。ただし、検証における意思決定間隔は  $t_{k+1} - t_k = 0.5$  (s) に変更された。以降、4コーナーDP試験の結果を5.5.3.1節で、着桟操船軌道の追従結果を5.4.3.1節で述べる。

#### 5.5.3.1 4コーナーDP試験結果

4コーナーDP試験の結果を示す。4コーナーDP試験結果は、DPSの評価のためによく実施される試験であり [91, 93], surge, sway, yaw のそれぞれ独立した運動に加えて、それらの連成運動における追従性能を示

Table 5.8: Hyperparameters for RL training.

$\Delta t = 0.5$ (s)	$\sigma_{x_0}^{(\text{obs})} = 0.03$ (m)	$e_0^{(\text{tol})} = 2L_{\text{ship}}$ (m)	$R_P = 10^{-6}$
$t_{k+1} - t_k = 5.0$ (s)	$\sigma_{y_0}^{(\text{obs})} = 0.03$ (m)	$e_\infty^{(\text{tol})} = B_{\text{ship}}/2$ (m)	$R_P = 10^{-6}$
$\gamma = 0.99$	$\sigma_\psi^{(\text{obs})} = 0.1$ (deg.)	$\lambda_e = 0.01$	$R_P = 10^{-4}$
$T_1 = 5.0$ (s)	$\sigma_u^{(\text{obs})} = 0.02$ (m/s)	$\sigma_u^{(\text{sys})} = 0.0$ (m/s)	
$T_2 = 10.0$ (s)	$\sigma_{v_m}^{(\text{obs})} = 0.02$ (m/s)	$\sigma_{v_m}^{(\text{sys})} = 0.0$ (m/s)	
$T_3 = 20.0$ (s)	$\sigma_r^{(\text{obs})} = 0.1$ (deg./s)	$\sigma_r^{(\text{sys})} = 0.0$ (deg./s)	
$T_4 = 40.0$ (s)			

Table 5.9: Initial state of maneuvering simulation environments.

Item	Definition
$x_0^{(\text{des})}$ (m)	is set to 0.0
$y_0^{(\text{des})}$ (m)	is set to 0.0
$\psi^{(\text{des})}$ (deg.)	is set to 0.0
$u^{(\text{des})}$ (m/s)	is randomly selected from $\mathcal{D}^{(\text{traj})}$
$v_m^{(\text{des})}$ (m/s)	is randomly selected from $\mathcal{D}^{(\text{traj})}$
$r^{(\text{des})}$ (deg./s)	is randomly selected from $\mathcal{D}^{(\text{traj})}$
$x_0$ (m)	is given by uniform distribution whose interval is $[x_0^{(\text{des})} - L_{\text{ship}}, x_0^{(\text{des})} + L_{\text{ship}}]$
$y_0$ (m)	is given by uniform distribution whose interval is $[y_0^{(\text{des})} - L_{\text{ship}}, y_0^{(\text{des})} + L_{\text{ship}}]$
$\psi$ (deg.)	is given by uniform distribution whose interval is $[\psi^{(\text{des})} - 10, \psi^{(\text{des})} + 10]$
$u$ (m/s)	is equal to $u^{(\text{des})}$
$v_m$ (m/s)	is equal to $v_m^{(\text{des})}$
$r$ (deg./s)	is equal to $r^{(\text{des})}$
$\bar{U}_T$ (m)	is randomly selected from $\mathcal{D}^{(\text{traj})}$
$\bar{\xi}_T$ (deg.)	is randomly selected from $\mathcal{D}^{(\text{traj})}$

すことが可能である。本研究で実施された 4 コーナー DP 試験は以下に述べる 5 ステップで構成される。

1.  $\psi = 0$  (deg.) の停止状態から、 $x_0$  軸の正の方向に 5 (m) 移動する。
2. 方位角を保ったまま ( $\psi = 0$  (deg.)),  $y_0$  軸の正の方向に 5 (m) 移動する。
3. 位置を保ったまま、方位角を時計回りに 45 (deg.) だけ変針させる。
4. 方位角  $\psi = 45$  (deg.) を保ったまま、 $x_0$  軸の負の方向に 5 (m) 移動する。
5. 方位角を反時計回りに 45 (deg.) だけ変針させつつ、並行して  $y_0$  軸の負の方向に 5 (m) 移動する。

これらの各 step の時間は 300 (s) である。これらの参照姿勢は離散的に与えられているため、本実験では Reference Fileter [58] を使用して、中間の連続的な参照姿勢変数ベクトル  $\eta^{(\text{des})}$  を生成した。各ステップで

Table 5.10: Used Layers for networks.

	$W_l$	$b_l$	$g$		$W_l$	$b_l$	$g$
$\mathbf{h}_1^{(fcl)}$	$\mathbb{R}^{256 \times 23}$	$\mathbb{R}^{256}$	ReLU	$\mathbf{h}_1^{(fcl)}$	$\mathbb{R}^{256 \times 26}$	$\mathbb{R}^{256}$	ReLU
$\mathbf{h}_2^{(fcl)}$	$\mathbb{R}^{256 \times 256}$	$\mathbb{R}^{256}$	ReLU	$\mathbf{h}_3^{(fcl)}$	$\mathbb{R}^{256 \times 256}$	$\mathbb{R}^{256}$	ReLU
$\mathbf{h}_3^{(fcl)}$	$\mathbb{R}^{3 \times 256}$	$\mathbb{R}^3$	Sigmoid	$\mathbf{h}_4^{(fcl)}$	$\mathbb{R}^{1 \times 256}$	$\mathbb{R}^1$	None

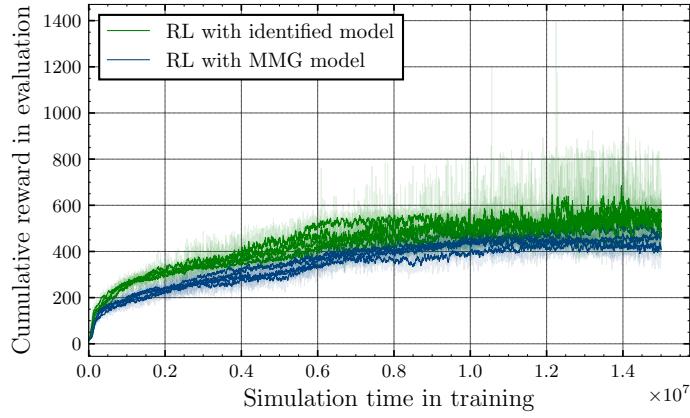


Fig. 5.19: Average cumulative reward of the 10 episodes used for the evaluation (Exponential Moving Average Value).

あたえられる参照姿勢を  $\boldsymbol{\eta}^{(\text{ref})}$  と表記すると、参照姿勢変数ベクトル  $\boldsymbol{\eta}^{(\text{des})}$  は微分方程式

$$\begin{pmatrix} \dot{\boldsymbol{\eta}}^{(\text{des})} \\ \ddot{\boldsymbol{\eta}}^{(\text{des})} \end{pmatrix} = \begin{bmatrix} 0 & \mathbf{I} & 0 \\ 0 & 0 & \mathbf{I} \\ -\boldsymbol{\Omega}^3 & -(2\Delta + \mathbf{I})\boldsymbol{\Omega}^2 & -(2\Delta + \mathbf{I})\boldsymbol{\Omega} \end{bmatrix} \begin{pmatrix} \boldsymbol{\eta}^{(\text{des})} \\ \dot{\boldsymbol{\eta}}^{(\text{des})} \\ \ddot{\boldsymbol{\eta}}^{(\text{des})} \end{pmatrix} + \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \boldsymbol{\Omega}^3 \end{bmatrix} \boldsymbol{\eta}^{(\text{ref})}, \quad (5.5.3)$$

を解くことで生成される。ここで、 $\mathbf{I} = \text{diag}(1.0, 1.0, 1.0)$  は 3 次元単位行列であり、 $\Delta = \mathbf{I}$  と  $\boldsymbol{\Omega} = \text{diag}(0.03, 0.03, 0.03)$  はそれぞれ減衰係数と固有周波数を表す。これらのパラメータは参照姿勢変数ベクトル  $\boldsymbol{\eta}^{(\text{des})}$  が実現可能なものとなるよう試行錯誤により決定した。

獲得された 5 つの最適制御方策のうちの 1 つを用いた 4 コーナー DP 試験の結果を Fig. 5.20 に示す。この結果から、対象船舶の発生させられるスラストが小さいため、軌道追従のために長い時間を必要とするものの、提案手法により得られた最適制御方策は風外乱の下において、surge, sway, yaw の独立運動および連成運動の追従を可能にしていることがわかる。また、得られた 5 つの最適制御方策のそれぞれの追従誤差を Fig. 5.21 に示す。提案手法の追従誤差のばらつきはほとんどの時間で大きな違いはないが、5 番目のステップで比較的大きな誤差を発生させた最適制御方策が存在することがわかる。また、MMG モデルを用いて訓練された最適制御方策と比較すると、追従誤差の大きさに大きな違いはないが、提案手法で観察された比較的大きなエラーは発生しなかった。この違いは操縦モデルの推定の有無による推測されるが、ほとんどの場合、操縦モデルの同定に起因するばらつきの影響は大きくないことを確認した。

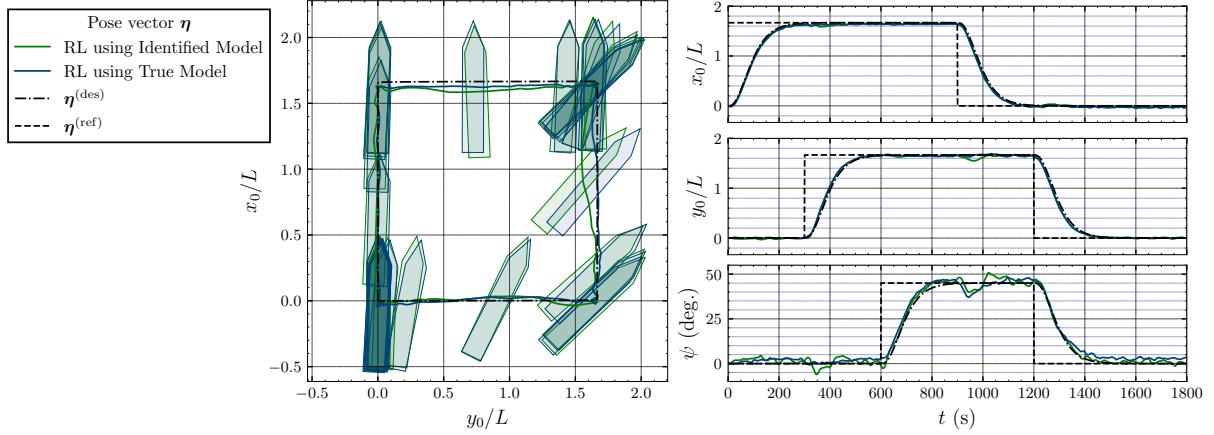


Fig. 5.20: Tracking results of the four corner DP test under wind disturbance,  $\bar{U}_T = 0.5$  (m/s),  $\bar{\xi}_T = 90$  (deg.), using an optimal control policy obtained by the proposed method.

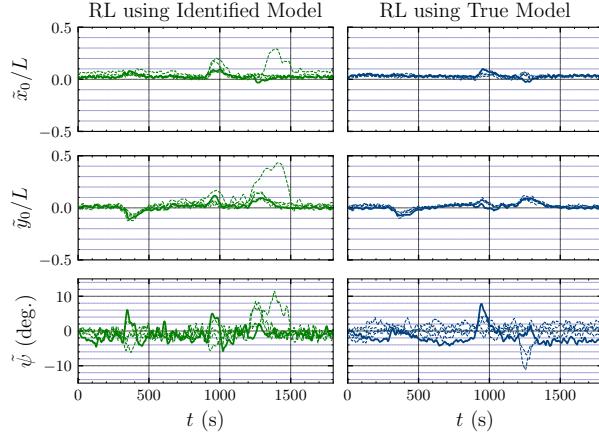


Fig. 5.21: Tracking error for the 4-corner DP test for each of the five different optimal policies obtained by the proposed method. The solid line represents the result of Fig. 5.20 and the dashed line represents the result using the other four policies.

### 5.5.3.2 着桟操船軌道データの追従結果

着桟操船軌道の追従実験結果を示す。ここで用いられた着桟操船軌道はフルケール版の対象船の着桟のための港湾内ナビゲーションデータから得られ、相似則により模型スケールへ変換された軌道である。使用された港湾内ナビゲーションデータに関する詳細は文献 [126] を参照されたい。

獲得された5つの最適制御方策のうちの1つを用いて行った着桟操船軌道の追従試験結果をFig. 5.22に示す。提案手法により得られた最適制御方策は参照軌道に船を追従させることができていることがわかる。ドッキング軌道は、船速が約0.4 (m/s) から約0.0 (m/s) まで低下しているが、制御方策はそれぞれの速度域で適切に追従可能であるように訓練されていることがわかる。例えば、船速が約0.4 (m/s) の時は二つの舵が0 (deg.) に近い角度で制御されており、一方で、船速が約0.0 (m/s) の時は二つの舵が、Hover角と呼ばれるプロペラが正回転していても全体のスラストが発生しない舵角に近い角度で制御されており、訓練による経験

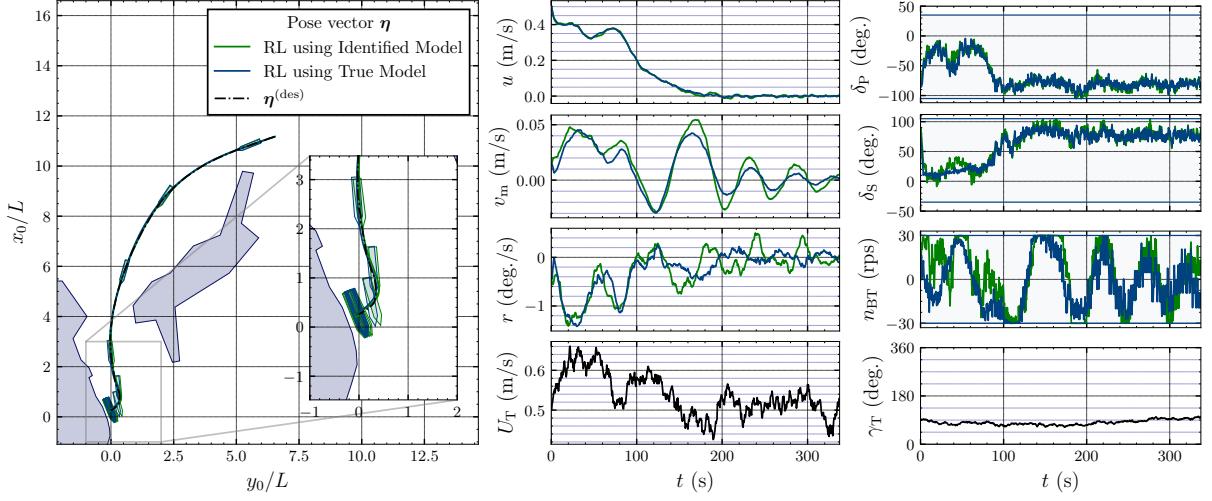


Fig. 5.22: Tracking results of the port navigation data under wind disturbance,  $\bar{U}_T = 0.5$  (m/s),  $\bar{\xi}_T = 90$  (deg.), using an optimal control policy obtained by the proposed method.

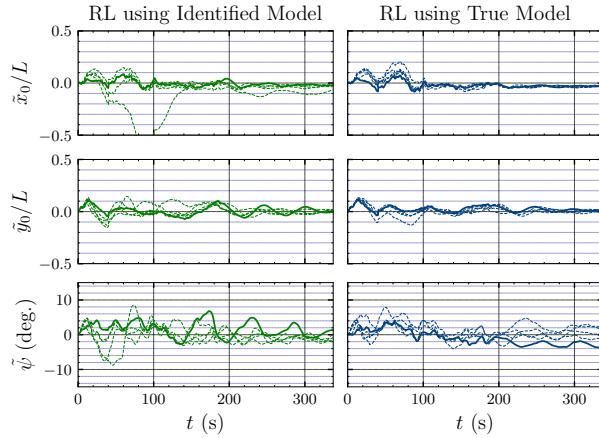


Fig. 5.23: Tracking error of the port navigation data for each of the five different optimal policies obtained by the proposed method. The solid line represents the result of Fig. 5.20 and the dashed line represents the result using the other four policies.

から適切な操船制御方策が獲得されている。

ランダムシードの異なる5つの最適制御方策を用いた着桟操船軌道の追従試験の追従誤差をFig. 5.23に示す。提案手法の追従誤差のばらつきについて、ドッキング軌道のアプローチフェーズ( $0\text{ (s)} < t < 150\text{ (s)}$ )では、一つのケースで約 $0.5L$ の比較的大きな誤差が発生している。しかし、この誤差は次第に減少しており、また、他のケースと時間では追従誤差の結果が大きくばらつくことがないことを確認した。また、MMG モデルを用いて訓練された最適制御方策と比較すると、4 コーナー DP 試験結果と同様に、追従誤差の大きさに大きな違いはないが、提案手法で観察された比較的大きなエラーは発生しなかった。しかし、ほとんどの場合、操縦モデルの同定に起因するばらつきの影響は大きくないことを確認した。

## 5.6 議論

5.4 節及び5.5 節では、提案手法による軌道追従制御方策の訓練結果および軌道追従結果を示した。それらの結果から判明した点を以下にまとめる。

- 5.4 節及び5.5 節のいずれの実験においても得られた軌道追従制御方策が着桟操船軌道を追従可能であることを示し、5.5 節では 4 コーナー DP 試験を実現可能であることを示した。従って、訓練環境設定、報酬関数設計、参照軌道のランダム生成手法を含む提案手法が、高速領域と低速領域の両方を含む着桟操船軌道に対応した軌道追従制御方策の獲得に有効であることが確認できた。特に、参照軌道のランダム生成手法は、制御方策が着桟操船軌道の追従で遭遇する可能性のある状態パターンを網羅的に経験させ、着桟操船軌道で直接訓練することなく着桟操船軌道の追従を可能にする。この手法は、5.2.2.2 節で述べたようにアクチュエータ構成に応じて制御入力のランダム生成手法が適切に決定される必要があるが、異なるアクチュエータ構成や船舶にも適用することが可能である。
- 5.4 節では、軌道追従制御方策に静的擬似障害物を考慮させることで、着桟操船軌道の追従実験における衝突確率が減少することを示した。よって、提案する報酬関数と静的擬似障害物の生成方法が、衝突を引き起こす可能性のある追従誤差を低減するのに有効であったことがわかる。しかし、得られた衝突確率は実用化するには依然として高い。そのため、風やモデル化誤差などの不確実性を考慮し、追従性能の向上が必要であり、また、フェンダー等の影響を考慮し、着桟操船の成功の定義を明確にした上で、より詳細な検討が必要である。
- 5.4 節では、MMG モデルで構成される操縦シミュレーション環境によって得られた軌道追従制御方策が、物理環境における模型実験でも着桟操船軌道を追従できることを示した。使用した MMG モデルは VecTwin 舵によって誘起される流体力などが一部簡略化されている。しかし、軌道追従制御方策はフィードバック制御を行うことに加えて、訓練ではシステムノイズが導入されているため、得られた軌道追従制御方策が一定のモデル化誤差を吸収可能であったと考えられる。
- 5.5 節では、操縦運動データ集合  $\mathcal{D}^{(traj)}$  を用いて得られた操縦モデルで構成される操縦シミュレーション環境によって得られた軌道追従制御方策が 4 コーナー DP 試験や着桟操船軌道の追従が可能であることを示した。よって、操縦運動データ集合が与えられるならば第 3 章で提案されたシステム同定手法を利用可能であり、必ずしも高精度な MMG モデルを用意する必要はない。しかし、3.4 節で述べたように操縦モデルの推定精度は与えられる操縦運動データに大きく依存し、操縦モデルの推定精度は軌道追従制御方策の追従精度に直結することから、提案手法は、与えられた操縦運動データ集合の分布外の状態における性能が低下する可能性を含んでおり、目的に対して十分に広く分布した操縦運動データ集合が必要となる。

以上のように、いくつかの限界が残されているものの、提案手法は MMG モデルのような操縦モデルもしくは操縦運動データ集合が与えられれば、高速領域と低速領域の両方を含む着桟操船軌道の追従制御方策を獲得することが可能である。実用上、風外乱は制御できないが、風外乱の力がアクチュエータの発生できる力の最大値を上回ると、アクチュエータをどのように制御しても接岸操作は困難になる。このような状況におけるアプローチも今後は検討する必要がある。

## 5.7 結言

本章では、強化学習を用いた軌道追従のための制御方策の獲得方法を提案した。提案手法では、まず、操縦モデル、アクチュエータ応答モデル、風の確率過程モデルで構成される操縦シミュレーション環境を構築される。そして、軌道追従のための制御方策がその操縦シミュレーション環境を用いた強化学習により獲得される。この訓練では、着棧操船軌道のような高速領域と低速領域の両方を含む軌道を追従可能な制御方策を獲得するために、ランダムに生成された参照軌道が用いられる。また、静的障害物との衝突を引き起こす追従誤差を優先的に回避するため、擬似障害物の自動生成を行い、制御方策に障害物との距離に関するペナルティが考慮される。提案手法は、操縦モデルもしくは操縦モデル推定のための操縦運動データ集合が与えられれば、着棧操船軌道のデータや障害物の地形データなどは不要で、また、物理環境における試行錯誤は行われない。シミュレーションおよび模型船を用いた実験では、提案手法により得られた軌道追従のための制御方策は着棧操船軌道の追従や4コーナーDP試験を実現可能であることが示された。また、提案手法で制御方策が静的障害物を考慮することで着棧操船軌道の追従実験における衝突確率を低下させることができることを示した。

## 第 6 章

### 総括

最後に、本学位論文の総括を述べる。本研究では港内操船および着桟操船の自動化に向けた操縦モデルおよび軌道追従制御に関する研究に取り組み、第 3 章では ANN を用いた港内操船および着桟操船のための操縦モデルの推定手法、第 4 章では ANN を用いた操縦モデル推定のためのデータ拡張手法、第 5 章では RL を用いた軌道追従制御方策の獲得手法について説明した。

第 3 章では、模型船の自由航走試験データを用いた実験により、(i) 加速度の計測値を必要としない軌道推定ベースの訓練手法は従来の回帰分析ベースの訓練手法と比較して RNN を用いた操縦モデルの推定精度を向上させること、(ii) ランダム操船試験を操縦モデルの推定へ用いることで低速操縦運動を含む着桟操船軌道の予測精度が向上すること、(iii) より長い期間の履歴影響を考慮した無限インパルス応答型 RNN が必ずしも操縦モデルの推定精度を向上させるわけではないこと、(iv) RNN を用いた操縦モデルは、分布が近い訓練データが与えられれば、拘束模型試験や経験式から得られた MMG モデルと同等かそれ以上に高い精度で低速操縦運動を予測可能であることが示された。ただし、実験では、最小でも実船スケール換算で約 7 時間程度のデータが使用されており、提案手法は十分な量の操縦運動データを必要とするに注意が必要である。

しかし、実船データ収集はコストが高いため、実用的には使用可能な操縦運動データ量が限られる可能性が高い。そのため、第 4 章では、操縦運動データ量が限られたとしても操縦モデルの推定精度を向上させるため、第 3 章で述べた操縦モデルの推定手法にスライシングとジッタリングと呼ばれる二つのデータ拡張手法を適用にも取り組んだ。模型船の自由航走試験データを用いた実験から、(v) 操縦運動データ量が限られている場合、スライシングとジッタリングは操縦モデルの推定精度を向上させる可能性のある有効なデータ拡張手法であること、(vi) スライシングとジッタリングはデータ集合の外挿領域の入力に対する操縦モデルの予測精度は改善できないことが示された。したがって、限界はあるもののデータ拡張手法により操縦モデルの推定精度を向上させることができ、さらに予測精度を向上させるためにはデータの分布が広く分散していく、外挿が少ないデータを用意することが好ましいと考えられる。

また、第 5 章では、軌道追従制御方策の獲得手法を提案している。提案手法では、操縦モデル、アクチュエータ応答モデル、風の確率過程モデルで構成される操縦シミュレーション環境を構築し、その操縦シミュレーション環境で軌道追従のための制御方策がモデルフリー強化学習により獲得される。シミュレーションと模型実験から、(vii) 訓練環境設定、報酬関数設計、参照軌道のランダム生成手法を含む提案手法で訓練された軌道追従制御方策は着桟操船軌道のような高速領域と低速領域の両方を含む軌道を追従可能であること、(viii) 軌道追従制御方策に静的障害物を考慮することで着桟操船軌道の追従実験における衝突確率が減少すること、(ix) MMG モデルを用いて得られた軌道追従制御方策が物理環境における模型実験でも着桟操船軌道を追従できること、(x) 操縦運動データ集合を用いて得られた操縦モデルを用いて得られた軌道追従制御方策が 4 コ

ーナー DP 試験や着桟操船軌道の追従が可能であることが示された。

上記の通り、本研究では、ANN を用いた操縦モデルの推定精度向上のための手法改善に取り組み、適切な操縦運動データ集合が利用可能であれば、標準船速と低船速の両方を含む操縦運動を表現する操縦モデルを獲得可能であることを示した。また、港内操船および着桟操船のための RL を用いた軌道追従制御方策の獲得手法を提案し、標準船速と低船速の両方を含む軌道を追従可能な制御方策を獲得可能であることを示した。さらに、二つの手法を組み合わせることで、操縦運動データから軌道追従制御方策を得られる可能性を示した。したがって、更なる検証が必要ではあるものの、本研究で提案する手法は、物理環境で計測された操縦運動データが与えられれば、どのようなアクチュエータ構成を持つ船舶であったとしても、港内操船および着桟操船のための操縦モデルと軌道追従制御方策を可能することが可能となった。しかし、いまだ操縦モデルの推定精度は与えられる操縦運動データの分布に依存するため、港内操船および着桟操船の自動化に向け、操縦モデルの予測の不確実性の定量化などに取り組むことが期待される。

## 参考文献

- [1] Yoshiki Miyauchi. *Optimizing Dynamic Model and Reference Trajectory on Berthing and Unberthing for a Maritime Autonomous Surface Ship*. PhD thesis, Osaka University, 3 2023. eng.
- [2] A Ogawa and H Kasai. On the mathematical model of manoeuvring motion of ships. *International Shipbuilding Progress*, Vol. 25, pp. 306–319, 1978.
- [3] H Yasukawa and Y Yoshimura. Introduction of mmg standard method for ship maneuvering predictions. *Journal of Marine Science and Technology*, Vol. 20, No. 1, pp. 37–52, 2015.
- [4] Kuniji Kose, Hiroyoshi Hinata, Yasuhisa Hashizume, and Eijiro Futagawa. On a mathematical model of maneuvering motions of ships in low speeds. *Journal of the Society of Naval Architects of Japan*, Vol. 1984, No. 155, pp. 132–138, 1984.
- [5] Masataka Fujino, Hiroshi Kagemoto, Yuji Ishii, and Hiroaki Joraku. Stopping ability of a ship in shallow water. *Journal of the Society of Naval Architects of Japan*, Vol. 1990, No. 168, pp. 117–129, 1990.
- [6] Hiroaki Kobayashi, Atsushi Ishibashi, Kenji Shimokawa, and Yuichi Shimura. A study on mathematical model for the maneuvering motions of twin-propeller twin-rudder ship : In reference to the maneuvering motion from ordinary speed range to low speed range. *The Journal of Japan Institute of Navigation*, Vol. 91, pp. 263–270, 1994.
- [7] Atsushi Ishibashi, Hiroaki Kobayashi, and Teruo Ugajin. A study on ship maneuvering characteristics in shallow water : On the harbor maneuvering at low speed range. *The Journal of Japan Institute of Navigation*, Vol. 95, pp. 371–380, 1996.
- [8] Yasuo Yoshimura, Ika Nakao, and Atsushi Ishibashi. Unified Mathematical Model for Ocean and Harbour Manoeuvring. In *Proceedings of MARSIM2009*, pp. 116–124. International Conference on Marine Simulation and Ship Maneuverability, aug 2009.
- [9] M S Chislett and J Strom-Tjensen. Planar motion mechanism tests and full-scale steering and manoeuvring predictions for a Mariner class vessel. *International Shipbuilding Progress*, Vol. 12, pp. 201–224, 1965.
- [10] Takeo Koyama, Jyong Hoe Chyu, Seizo Motora, and Masashiro Koyanagi. On the circular motion test technique (cmt) for the maneuverability model test. *Journal of the Society of Naval Architects of Japan*, Vol. 1975, No. 138, pp. 151–157, 1975.
- [11] Seizo Motora. On the measurement of added mass and added moment of inertia for ship motions. *Journal of Zosen Kikai*, Vol. 1959, No. 105, pp. 83–92, 1959.
- [12] S. Inoue, M. Hirano, and K. Kijima. Hydrodynamic derivatives on ship manoeuvring. *International*

- Shipbuilding Progress*, Vol. 28, pp. 112–125, 1 1981.
- [13] Katsuro Kijima, Toshiyuki Katsuno, Yasuaki Nakiri, and Yoshitaka Furukawa. On the manoeuvring performance of a ship with the parameter of loading condition. *Journal of the Society of Naval Architects of Japan*, Vol. 1990, No. 168, pp. 141–148, 1990.
  - [14] Hitoshi Fujii and Tatuo Tuda. Experimental researches on rudder performance. (2). *Journal of Zosen Kiokai*, Vol. 1961, No. 110, pp. 31–42, 1961.
  - [15] Toshifumi Fujiwara, Michio Ueno, and Tadashi Nimura. Estimation of Wind Forces and Moments acting on Ships. *Journal of the Society of Naval Architects of Japan*, Vol. 1998, No. 183, pp. 77–90, 1998.
  - [16] K.J. Åström and C.G. Källström. Identification of ship steering dynamics. *Automatica*, Vol. 12, No. 1, pp. 9–22, 1976.
  - [17] Martin A Abkowitz. Measurement of hydrodynamic characteristics from ship maneuvering trials by system identification. In *Transactions of Society of Naval Architects and Marine Engineers 88*, pp. 283–318, 1980.
  - [18] C.G. Källström and K.J. Åström. Experiences of system identification applied to ship steering. *Automatica*, Vol. 17, No. 1, pp. 187–198, 1981.
  - [19] Hyeon Kyu Yoon and Key Pyo Rhee. Identification of hydrodynamic coefficients in ship maneuvering equations of motion by estimation-before-modeling technique. *Ocean Engineering*, Vol. 30, No. 18, pp. 2379–2404, 2003.
  - [20] Serge Sutulo and C. Guedes Soares. An algorithm for offline identification of ship manoeuvring mathematical models from free-running tests. *Ocean Engineering*, Vol. 79, pp. 10–25, 2014.
  - [21] 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 (Japan)*, Vol. 26, pp. 541–554, 6 2021.
  - [22] Yoshiki Miyauchi, Atsuo Maki, Naoya Umeda, Dimas M Rachman, and Youhei Akimoto. System parameter exploration of ship maneuvering model for automatic docking/berthing using CMA-ES. *Journal of Marine Science and Technology*, Vol. 27, No. 2, pp. 1065–1083, 2022.
  - [23] Serge Sutulo and C. Guedes Soares. Application of an offline identification algorithm for adjusting parameters of a modular manoeuvring mathematical model. *Ocean Engineering*, Vol. 279, p. 114328, 2023.
  - [24] Yoshiki Miyauchi, Youhei Akimoto, and Atsuo Maki. Development of a mathematical model for harbor maneuvers to realize modeling automation. *Journal of Marine Science and Technology*, 2024.
  - [25] L. Moreira and C. Guedes Soares. Dynamic model of manoeuvrability using recursive neural networks. *Ocean Engineering*, Vol. 30, No. 13, pp. 1669 – 1697, 2003.
  - [26] Dmitry A. Osokin, Alexander A. Dyda, and Vasily E. Markin. Neural network identification of marine ship dynamics. *IFAC Proceedings Volumes*, Vol. 46, No. 33, pp. 191 – 196, 2013. 9th IFAC Conference on Control Applications in Marine Systems.
  - [27] Yifan Xue, Yanjun Liu, Chen Ji, Gang Xue, and Shuteng Huang. System identification of ship dynamic model based on gaussian process regression with input noise. *Ocean Engineering*, Vol.

- 216, p. 107862, 2020.
- [28] Kouki Wakita, Atsuo Maki, Naoya Umeda, Yoshiki Miyauchi, Tohga Shimoji, Dimas M Rachman, and Youhei Akimoto. On neural network identification for low-speed ship maneuvering model. *Journal of Marine Science and Technology*, Vol. 27, No. 1, pp. 772–785, 2022.
- [29] W. L. Luo and Z. J. Zou. Parametric Identification of Ship Maneuvering Models by Using Support Vector Machines. *Journal of Ship Research*, Vol. 53, No. 01, pp. 19–30, 03 2009.
- [30] Xin-guang Zhang and Zao-jian Zou. Identification of Abkowitz Model for Ship Manoeuvring Motion Using  $\epsilon$ -Support Vector Regression. *Journal of Hydrodynamics*, Vol. 23, No. 3, pp. 353–360, 2011.
- [31] Weilin Luo, Lúcia Moreira, and C. Guedes Soares. Manoeuvring simulation of catamaran by using implicit models based on support vector machines. *Ocean Engineering*, Vol. 82, pp. 150–159, 2014.
- [32] Wenhe Shen, Jianxi Yao, Xinjue Hu, Jialun Liu, and Shijie Li. Ship dynamics model identification based on semblance least square support vector machine. *Ocean Engineering*, Vol. 287, p. 115908, 2023.
- [33] Kouki Wakita, Yoshiki Miyauchi, Youhei Akimoto, and Atsuo Maki. Data augmentation methods of parameter identification of a dynamic model for harbor maneuvers, 2023.
- [34] Man Zhu, Axel Hahn, Yuan-Qiao Wen, and Wu-Qiang Sun. Optimized support vector regression algorithm-based modeling of ship dynamics. *Applied Ocean Research*, Vol. 90, p. 101842, 2019.
- [35] Zihao Wang, Zaojian Zou, and C. Guedes Soares. Identification of ship manoeuvring motion based on nu-support vector machine. *Ocean Engineering*, Vol. 183, pp. 270–281, 2019.
- [36] Rin Suyama, Rintaro Matsushita, Ryo Kakuta, Kouki Wakita, and Atsuo Maki. Parameter fine-tuning method for mmg model using real-scale ship data. *Ocean Engineering*, Vol. 298, p. 117323, 2024.
- [37] Zihao Wang, Haitong Xu, Li Xia, Zaojian Zou, and C. Guedes Soares. Kernel-based support vector regression for nonparametric modeling of ship maneuvering motion. *Ocean Engineering*, Vol. 216, p. 107994, 2020.
- [38] Hong-Wei He, Zi-Hao Wang, Zao-Jian Zou, and Yi Liu. Nonparametric modeling of ship maneuvering motion based on self-designed fully connected neural network. *Ocean Engineering*, Vol. 251, p. 111113, 5 2022.
- [39] Lijia Chen, Peiyi Yang, Shengwei Li, Yanfei Tian, Guangqiang Liu, and Guozhu Hao. Grey-box identification modeling of ship maneuvering motion based on ls-svm. *Ocean Engineering*, Vol. 266, p. 112957, 2022.
- [40] G Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals and Systems*, Vol. 2, No. 4, pp. 303–314, 1989.
- [41] Kurt Hornik. Approximation capabilities of multilayer feedforward networks. *Neural Networks*, Vol. 4, No. 2, pp. 251 – 257, 1991.
- [42] L Moreira and C Guedes Soares. Recursive neural network model of catamaran manoeuvring. *International Journal of Maritime Engineering*, Vol. 154, , 2012. Copyright The University of Buckingham Press 2012.
- [43] G. Rajesh and S.K. Bhattacharyya. System identification for nonlinear maneuvering of large tankers using artificial neural network. *Applied Ocean Research*, Vol. 30, No. 4, pp. 256–263, 2008.

- [44] Xin-Guang Zhang and Zao-Jian Zou. Black-box modeling of ship manoeuvring motion based on feed-forward neural network with chebyshev orthogonal basis function. *Journal of Marine Science and Technology*, Vol. 18, pp. 42–49, 3 2013.
- [45] Lizhu Hao, Yang Han, Chao Shi, and Ziying Pan. Recurrent neural networks for nonparametric modeling of ship maneuvering motion. *International Journal of Naval Architecture and Ocean Engineering*, Vol. 14, p. 100436, 1 2022.
- [46] Joohyun Woo, Jongyoung Park, Chanwoo Yu, and Nakwan Kim. Dynamic model identification of unmanned surface vehicles using deep learning network. *Applied Ocean Research*, Vol. 78, pp. 123–133, 2018.
- [47] Yan Jiang, Xian-Rui Hou, Xue-Gang Wang, Zi-Hao Wang, Zhao-Long Yang, and Zao-Jian Zou. Identification modeling and prediction of ship maneuvering motion based on lstm deep neural network. *Journal of Marine Science and Technology*, Vol. 27, pp. 125–137, 3 2022.
- [48] Wilmer Ariza Ramirez, Zhi Quan Leong, Hung Nguyen, and Shantha Gamini Jayasinghe. Non-parametric dynamic system identification of ships using multi-output gaussian processes. *Ocean Engineering*, Vol. 166, pp. 26–36, 2018.
- [49] Yifan Xue, Gang Chen, Zhitong Li, Gang Xue, Wei Wang, and Yanjun Liu. Online identification of a ship maneuvering model using a fast noisy input gaussian process. *Ocean Engineering*, Vol. 250, p. 110704, 2022.
- [50] Kouichi Shouji and Kohei Ohtsu. A study on the optimization of ship maneuvering by optimal control theory (1st report). *Journal of the Society of Naval Architects of Japan*, Vol. 1992, No. 172, pp. 365–373, 1992.
- [51] Glenn Bitar, Andreas B. Martinsen, Anastasios M. Lekkas, and Morten Breivik. Trajectory planning and control for automatic docking of asvs with full-scale experiments. *IFAC-PapersOnLine*, Vol. 53, No. 2, pp. 14488–14494, 2020. 21st IFAC World Congress.
- [52] Andreas B. Martinsen, Glenn Bitar, Anastasios M. Lekkas, and SÃ©bastien Gros. Optimization-based automatic docking and berthing of asvs using exteroceptive sensors: Theory and experiments. *IEEE Access*, Vol. 8, pp. 204974–204986, 2020.
- [53] Yaseen Adnan Ahmed and Kazuhiko Hasegawa. Consistently trained artificial neural network for automatic ship berthing control. *TransNav, the International Journal on Marine Navigation and Safety of Sea Transportation*, Vol. 9, No. 3, pp. 417–426, 2015.
- [54] 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 (Japan)*, Vol. 28, pp. 452–470, 6 2023.
- [55] Eivind Meyer, Amalie Heiberg, Adil Rasheed, and Omer San. Colreg-compliant collision avoidance for unmanned surface vehicle using deep reinforcement learning. *IEEE Access*, Vol. 8, pp. 165344–165364, 2020.
- [56] 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 and Technology*, Vol. 26, pp. 509–524, 6 2021.
- [57] Asgeir J. SÃ rensen. A survey of dynamic positioning control systems. *Annual Reviews in Control*,

- Vol. 35, pp. 123–136, 4 2011.
- [58] Thor I. Fossen. *Handbook of Marine Craft Hydrodynamics and Motion Control*. Wiley, 4 2011. Wiley Online Books.
- [59] Simon J. N. Lexau, Morten Breivik, and Anastasios M. Lekkas. Automated docking for marine surface vessels—a survey. *IEEE Access*, Vol. 11, pp. 132324–132367, 2023.
- [60] 小山健夫, 野本謙作. 自動操縦について—海上交通の制御一. 第2回操縦性シンポジウム, 1970.
- [61] 小瀬邦治, 福戸淳司, 菅野賢治, 赤木茂, 原田美秀子. 船の自動離着棧システムに関する研究. 日本造船学会論文集, Vol. 1986, No. 160, pp. 103–110, 1986.
- [62] 小山健夫, 金雁, 金奎煥. 船の自動離着棧のシステム的考察(第1報). 日本造船学会論文集, Vol. 1987, No. 162, pp. 201–210, 1987.
- [63] 吉久英昭. 高信頼度知能化船(高度自動運航システム)：(その3)自動離着棧システム. 日本造船学会誌, Vol. 722, No. 0, pp. 498–502, 1989.
- [64] 高井忠夫, 大津皓平. 汐路丸による自動離着棧実験について. 日本航海学会論文集, Vol. 83, No. 0, pp. 267–276, 1990.
- [65] 長谷川和彦, 山本郁夫, 正司公一. 港内操船への自動化への挑戦. 運動性能研究委員会第13回シンポジウム(船体運動及びその制御と海象), 1997.
- [66] Naoki Mizuno, Yosuke Uchida, and Tadatsugi Okazaki. Quasi real-time optimal control scheme for automatic berthing. *IFAC-PapersOnLine*, Vol. 48, No. 16, pp. 305–312, 2015. 10th IFAC Conference on Manoeuvring and Control of Marine Craft MCMC 2015.
- [67] Shijie Li, Jialun Liu, Rudy R. Negenborn, and Qing Wu. Automatic docking for underactuated ships based on multi-objective nonlinear model predictive control. *IEEE Access*, Vol. 8, pp. 70044–70057, 2020.
- [68] Dimas M. Rachman, Atsuo Maki, Yoshiki Miyauchi, and Naoya Umeda. Warm-started semionline trajectory planner for ship's automatic docking (berthing). *Ocean Engineering*, Vol. 252, p. 111127, 2022.
- [69] Namkyun Im, Seong-Keon Lee, and Do Bang Hyung. An application of ann to automatic ship berthing using selective controller. *TransNav, the International Journal on Marine Navigation and Safety of Sea Transportation*, Vol. 1, No. 1, pp. 101–105, 2007.
- [70] Nam-Kyun Im and Van-Suong Nguyen. Artificial neural network controller for automatic ship berthing using head-up coordinate system. *International Journal of Naval Architecture and Ocean Engineering*, Vol. 10, No. 3, pp. 235–249, 2018.
- [71] Yaseen Adnan Ahmed and Kazuhiko Hasegawa. Automatic ship berthing using artificial neural network trained by consistent teaching data using nonlinear programming method. *Engineering Applications of Artificial Intelligence*, Vol. 26, No. 10, pp. 2287–2304, 2013.
- [72] Yonghui Shuai, Guoyuan Li, Xu Cheng, Robert Skulstad, Jinshan Xu, Honghai Liu, and Houxiang Zhang. An efficient neural-network based approach to automatic ship docking. *Ocean Engineering*, Vol. 191, p. 106514, 2019.
- [73] Kouichi Shouji, Kohei Ohtsu, and Sumitoshi Mizoguchi. An automatic berthing study by optimal control techniques. *IFAC Proceedings Volumes*, Vol. 25, pp. 185–194, 4 1992.
- [74] Kouichi Shouji, Kohei Ohtsu, and Toshiyuki Hotta. A study on the optimization of ship maneu-

- vering by optimal control theory (2nd report). *Journal of the Society of Naval Architects of Japan*, Vol. 1993, No. 173, pp. 221–229, 1993.
- [75] Kouichi Shouji and Kohei Ohtsu. A study on the optimization of ship maneuvering by optimal control theory (3rd report). *Journal of the Society of Naval Architects of Japan*, Vol. 1993, No. 174, pp. 339–344, 1993.
- [76] A. K. Wu and A. Miele. Sequential conjugate gradient-restoration algorithm for optimal control problems with non-differential constraints and general boundary conditions, part i. *Optimal Control Applications and Methods*, Vol. 1, No. 1, pp. 69–88, 1980.
- [77] Andreas B. Martinsen, Anastasios M. Lekkas, and Sébastien Gros. Autonomous docking using direct optimal control. *IFAC-PapersOnLine*, Vol. 52, No. 21, pp. 97–102, 2019. 12th IFAC Conference on Control Applications in Marine Systems, Robotics, and Vehicles CAMS 2019.
- [78] 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*, Vol. 245, p. 110390, 2022.
- [79] Atsuo Maki, Naoki Sakamoto, Youhei Akimoto, Hiroyuki Nishikawa, and Naoya Umeda. Application of optimal control theory based on the evolution strategy (CMA-ES) to automatic berthing. *Journal of Marine Science and Technology*, Vol. 25, No. 1, pp. 221–233, 2020.
- [80] Atsuo Maki, Youhei Akimoto, and Umeda Naoya. Application of optimal control theory based on the evolution strategy (CMA-ES) to automatic berthing (part: 2). *Journal of Marine Science and Technology*, Vol. 26, No. 3, pp. 835–845, 2021.
- [81] Kinzo INOUE, Hiroaki SETA, and Kenji MASUDA. Guidelines for speed reduction in berthing manoeuvre. *The Journal of Japan Institute of Navigation*, Vol. 107, pp. 169–176, 2002.
- [82] Huarong Zheng, Rudy R. Negenborn, and Gabriel Lodewijks. Trajectory tracking of autonomous vessels using model predictive control. *IFAC Proceedings Volumes*, Vol. 47, No. 3, pp. 8812–8818, 2014.
- [83] Chenguang Liu, Huarong Zheng, Rudy R Negenborn, Xiumin Chu, and Le Wang. Trajectory tracking control for underactuated surface vessels based on nonlinear model predictive control. In Francesco Corman, Stefan Voß, and Rudy R Negenborn, editors, *Computational Logistics*, pp. 166–180. Springer International Publishing, 2015.
- [84] Fuguang Ding, Yuanhui Wang, and Yong Wang. Trajectory-tracking controller design of underactuated surface vessels. In *OCEANS'11 MTS/IEEE KONA*, pp. 1–5, 2011.
- [85] Yang Yang, Jialu Du, Hongbo Liu, Chen Guo, and Ajith Abraham. A trajectory tracking robust controller of surface vessels with disturbance uncertainties. *IEEE Transactions on Control Systems Technology*, Vol. 22, No. 4, pp. 1511–1518, 2014.
- [86] Guoxing Wen, Shuzhi Sam Ge, C. L. Philip Chen, Fangwen Tu, and Shengnan Wang. Adaptive tracking control of surface vessel using optimized backstepping technique. *IEEE Transactions on Cybernetics*, Vol. 49, No. 9, pp. 3420–3431, 2019.
- [87] Dimas M. Rachman, Yusuke Aoki, Yoshiki Miyauchi, Naoya Umeda, and Atsuo Maki. Experimental low-speed positioning system with vectwin rudder for automatic docking (berthing). *Journal of Marine Science and Technology*, Vol. 28, pp. 689–703, 9 2023.

- [88] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. The MIT Press, second edition, 2018.
- [89] Yin Cheng and Weidong Zhang. Concise deep reinforcement learning obstacle avoidance for underactuated unmanned marine vessels. *Neurocomputing*, Vol. 272, pp. 63–73, 1 2018.
- [90] Andreas B. Martinsen and Anastasios M. Lekkas. Curved path following with deep reinforcement learning: Results from three vessel models. *OCEANS 2018 MTS/IEEE Charleston, OCEAN 2018*, 2019.
- [91] Andreas B. Martinsen, Anastasios M. Lekkas, Sébastien Gros, Jon Arne Glomsrud, and Tom Arne Pedersen. Reinforcement learning-based tracking control of usvs in varying operational conditions. *Frontiers in Robotics and AI*, Vol. 7, , 2020.
- [92] Ning Wang, Ying Gao, Hong Zhao, and Choon Ki Ahn. Reinforcement learning-based optimal tracking control of an unknown unmanned surface vehicle. *IEEE Transactions on Neural Networks and Learning Systems*, Vol. 32, pp. 3034–3045, 7 2021.
- [93] Andreas B. Martinsen, Anastasios M. Lekkas, and Sébastien Gros. Reinforcement learning-based nmpc for tracking control of asvs: Theory and experiments. *Control Engineering Practice*, Vol. 120, p. 105024, 2022.
- [94] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin Riedmiller, Andreas K. Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. Human-level control through deep reinforcement learning. *Nature*, Vol. 518, pp. 529–533, 2 2015.
- [95] Andreas B. Martinsen and Anastasios M. Lekkas. Straight-path following for underactuated marine vessels using deep reinforcement learning. *IFAC-PapersOnLine*, Vol. 51, pp. 329–334, 2018.
- [96] Forng-Chen Chiu, Tun-Li Chang, Jenhwa Go, Shean-Kwang Chou, and Wei-Chung Chen. A recursive neural networks model for ship maneuverability prediction. In *Oceans '04 MTS/IEEE Techno-Ocean '04 (IEEE Cat. No.04CH37600)*, Vol. 3, pp. 1211–1218 Vol.3, Nov 2004.
- [97] Qingsong Wen, Liang Sun, Fan Yang, Xiaomin Song, Jingkun Gao, Xue Wang, and Huan Xu. Time series data augmentation for deep learning: A survey. In Zhi-Hua Zhou, editor, *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, pp. 4653–4660. International Joint Conferences on Artificial Intelligence Organization, 8 2021. Survey Track.
- [98] Brian Kenji Iwana and Seiichi Uchida. An empirical survey of data augmentation for time series classification with neural networks. *PLOS ONE*, Vol. 16, No. 7, pp. 1–32, 07 2021.
- [99] Arthur Le Guennec, Simon Malinowski, and Romain Tavenard. Data Augmentation for Time Series Classification using Convolutional Neural Networks. In *ECML/PKDD Workshop on Advanced Analytics and Learning on Temporal Data*, Riva Del Garda, Italy, September 2016.
- [100] Terry T. Um, Franz M. J. Pfister, Daniel Pichler, Satoshi Endo, Muriel Lang, Sandra Hirche, Urban Fietzek, and Dana Kulic. Data augmentation of wearable sensor data for parkinson's disease monitoring using convolutional neural networks. In *Proceedings of the 19th ACM International Conference on Multimodal Interaction*, ICMI '17, p. 216–220, New York, NY, USA, 2017. Association for Computing Machinery.

- [101] Hassan Ismail Fawaz, Germain Forestier, Jonathan Weber, Lhassane Idoumghar, and Pierre-Alain Muller. Data augmentation using synthetic data for time series classification with deep residual networks. In *International Workshop on Advanced Analytics and Learning on Temporal Data, ECML PKDD*, 2018.
- [102] Scott Fujimoto, Herke van Hoof, and David Meger. Addressing function approximation error in actor-critic methods. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, Vol. 80 of *Proceedings of Machine Learning Research*, pp. 1587–1596. PMLR, 10–15 Jul 2018.
- [103] Thomas M. Moerland, Joost Broekens, Aske Plaat, and Catholijn M. Jonker. Model-based reinforcement learning: A survey. *Foundations and Trends® in Machine Learning*, Vol. 16, pp. 1–118, 2023.
- [104] Kouki Wakita, Youhei Akimoto, Dimas M. Rachman, Yoshiki Miyuchi, and Atsuo Maki. Collision probability reduction method for tracking control in automatic docking/berthing using reinforcement learning. *Journal of Marine Science and Technology*, Vol. 28, pp. 844–861, 12 2023.
- [105] Kouki Wakita. Model-Based Reinforcement Learning for Trajectory Tracking Control of Autonomous Surface Ship. In *The 34th International Ocean and Polar Engineering Conference*, International Ocean and Polar Engineering Conference, pp. ISOPE-I-24-524, 06 2024.
- [106] Suisei Wada, Naoya Umeda, and Atsuo Maki. Development of general purpose free-running model ship with ros : Enhanced model ship experiments system. *Conference proceedings, the Japan Society of Naval Architects and Ocean Engineers (in Japanese)*, No. 28, pp. 587–594, jun 2019.
- [107] H. Kobayashi, J.J. J Blok, R. Barr, Y. S. Kim, and J. Nowicki. The Specialist Committee on Esso Osaka Final Report and Recommendations to the 23rd ITTC. *23rd International Towing Tank Conference*, Vol. II, pp. 581–743, 2002.
- [108] Hironori YASUKAWA and Kuniji KOSE. Simulation of Stopping Maneuver of a Tanker in Wind and Waves. In *Transactions of the West-Japan Society of Naval Architects*, Vol. 106, pp. 57–68, 2003.
- [109] Michio Ueno, Tadashi Nimura, Hideki Miyazaki, Toshifumi Fujiwara, Koji Nonaka, and Hideo Yabuki. Model Experiment and Sea Trial for Investigating Manoeuvrability of a Training Ship. *Journal of the Society of Naval Architects of Japan*, Vol. 2001, No. 189, pp. 71–80, 2001.
- [110] Kazuhiko Hasegawa and Takeshi Fukutomi. On Harbour Manoeuvring and Neural Control System for Berthing with Tug Operation. In *Proc. of 3rd International Conference Manoeuvring and Control of Marine Craft (MCMC'94)*, pp. pp.197–210, 1994.
- [111] Hitoshi FUJII and Tatuo TUDA. Experimental Researches on Rudder Performance. (2). *Journal of Zosen Kiokai*, Vol. 1961, No. 110, pp. 31–42, 1961.
- [112] Yasuo Yoshimura and Kensaku Nomoto. Modeling of manoeuvring behaviour of ships with a propeller idling, boosting and reversing. *Journal of the Society of Naval Architects of Japan*, Vol. 1978, No. 144, pp. 57–69, 1978.
- [113] Yasushi Kitagawa, Yoshiaki Tsukada, and Hideki Miyazaki. A Study on Mathematical Models of Propeller and Rudder under Maneuvering with Propeller Reverse Rotation. *Conference Proceedings The Japan Society of Naval Architects and Ocean Engineers*, Vol. 20, pp. 117–120, 2015.

- [114] Japan Hamworthy & Co., Ltd. Steering systems: New Vectwin system. [https://www.japanham.com/en/service/new\\_vectwin.html](https://www.japanham.com/en/service/new_vectwin.html), 2022. Accessed on 12 July 2023.
- [115] Kazuhiko Hasegawa, Donghoon Kang, Masaaki Sano, and Kenjiro Nabeshima. Study on the maneuverability of a large vessel installed with a mariner type super vectwin rudder. *Journal of Marine Science and Technology*, Vol. 11, pp. 88–99, 6 2006.
- [116] Hironori YASUKAWA, Noritaka HIRATA, Susumu TANAKA, and Hiroki HATA. Tank tests on low speed maneuvering of a ship with vectwin rudder. *The Journal of Japan Institute of Navigation*, Vol. 124, pp. 265–271, 2011.
- [117] Donghoon Kang, Vishwanath Nagarajan, Kazuhiko Hasegawa, and Masaaki Sano. Mathematical model of single-propeller twin-rudder ship. *Journal of Marine Science and Technology*, Vol. 13, pp. 207–222, 2008.
- [118] Eiichi Kobayashi. A simulation study on ship manoeuvrability at low speeds. Akishima Laboratory, Ocean Engineering Research Section, Mitsubishi Heave Industries Ltd. Published in: Mitsubishi Technical Bulletin No. 180, 1988.
- [119] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C.J. Burges, L. Bottou, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, Vol. 25. Curran Associates, Inc., 2012.
- [120] Geoffrey Hinton, Li Deng, Dong Yu, George E. Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara N. Sainath, and Brian Kingsbury. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Processing Magazine*, Vol. 29, No. 6, pp. 82–97, 2012.
- [121] Hiroaki NAKANISHI, Takehisa KOHDA, and Koichi INOUE. A design method of optimal state feed-back control systems by use of neural network. *Transactions of the Society of Instrument and Control Engineers*, Vol. 33, No. 9, pp. 882–889, 1997.
- [122] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.
- [123] Zhicheng Cui, Wenlin Chen, and Yixin Chen. Multi-scale convolutional neural networks for time series classification. 2016.
- [124] Chris M. Bishop. Training with Noise is Equivalent to Tikhonov Regularization. *Neural Computation*, Vol. 7, No. 1, pp. 108–116, 01 1995.
- [125] Khandakar M. Rashid and Joseph Louis. Times-series data augmentation and deep learning for construction equipment activity recognition. *Advanced Engineering Informatics*, Vol. 42, p. 100944, 2019.
- [126] Yoshiki Miyauchi, Taichi Kambara, Naoya Umeda, Kazuyoshi Hosogaya, and Atsuo Maki. Statistical analysis of port navigation and maneuver of a Japanese merchant vessel. In *Conference proceedings, the Japan Society of Naval Architects and Ocean Engineers*, Vol. 35, pp. 77–87, 2022. (in Japanese).
- [127] 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*, Vol. 27, pp. 1302–1313, 12 2022.
- [128] Atsuo Maki, Yuuki Maruyama, Leo Dostal, Kenji Sasa, Ryohei Sawada, and Kouki Wakita. Practical method for evaluating wind influence on autonomous ship operations (2nd report). *Journal of Marine Science and Technology*, 2024.
- [129] Naoki Sakamoto and Youhei Akimoto. Modified box constraint handling for the covariance matrix adaptation evolution strategy. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, GECCO '17, pp. 183–184, New York, NY, USA, 2017. Association for Computing Machinery.