深層強化学習を用いた複数の船舶を考慮した自動避航操船

澤田 涼平*

Automatic Collision Avoidance Maneuvers Considering Multiple Vessels' Encounter Using Deep Reinforcement Learning

by

SAWADA Ryohei

Abstract

This paper presents the algorithms of automatic collision avoidance for multiple vessels' encounter using deep reinforcement learning (DRL) . In order to assess the risk of collision, we used a method based on Obstacle Zone by Target (OZT), which expresses the areas where ships will collide with each other in the future. In the computation of OZT, we show how to take the bow crossing range (BCR) into account in order to accurately represent encounter situations. We also extended OZT with the inside OZT to stabilize the learning process of DRL. We developed a method to efficiently process OZT and represent the distribution of OZT as a single vector whose components are binary, and named it Grid Sensor. PPO, which is classified as an actor-critic method, was employed to learn collision avoidance of ships. The deep neural networks representing the actor and the critic were trained using a combination of convolutional and full-connected layers in the discrete action space and a combination of the long short-term memory (LSTM) cell in a continuous action space. The trained model has passed all scenarios of Imazu problem. The model is also validated by a test scenario which includes more ships than each scenario of Imazu problem.

＊　知識・データシステム系

原稿受付　令和 3 年 1 月 25 日
審 査 日　令和 3 年 3 月 12 日

目　次

1. まえがき …… 2
2. Obstacle Zone by Target（OZT） …… 3
2.1 OZT の計算法 …… 3
2.2 前方航過距離を考慮した OZT の計算 …… 4
2.3 内部 OZT …… 5
2.4 OZT の検知手法：グリッドセンサー …… 6
3. 深層強化学習による避航操船の学習 …… 6
3.1 深層強化学習 …… 7
3.1.1 深層強化学習アルゴリズムの実装 …… 7
3.1.2 ネットワークの設計と更新 …… 8
3.2 学習のための環境設計 …… 9
3.2.1 報酬の設計 …… 10
4. 学習モデルの検証 …… 11
4.1 今津問題による検証 …… 11
4.2 船舶 5 隻が航行するシナリオでの検証 …… 13
5. まとめ …… 14
謝辞 …… 15
参考文献 …… 15

1.　まえがき

自動運航船を実現する上で重要な要素技術の一つに船舶の衝突を回避する自動避航操船機能の実装がある．自動避航操船を実装するための手法は，これまでに数多く提案されている．一方で，既存の手法の多くは複数の船舶が航行するシナリオにおいて適切な避航操船を行うことができない，もしくは避航操船のアルゴリズムを適用する際に，避航判定の各試行で原理的に周囲に複数存在する船舶のうちたった一隻のみしか考慮に入れることができないことが指摘されていた [1]．複数の船舶を考慮した避航操船は，東京湾を始めとする輻輳海域においては特に重要となるが，その取り扱いにはいくつかの困難がある．まず制御対象となる自船のまわりを航行する船舶の数は不定であるため，ある固定隻数もしくは隻数の上限を設定した避航判定アルゴリズムでは，同時に考慮できる隻数が限られてしまうだけでなく，周囲に存在する複数存在する船のうちどの船を考慮に入れるか（自船からの距離や衝突リスクのようなある種の判定基準に基づく場合が多い）を設計する必要が生じる．例えば，自船と最も距離の近い船舶のみを逐次避航するといったアルゴリズムの場合，避航操船を行った先で別の船舶と新たに見合い関係が生じるといったことが繰り返し起きる可能性が高まり，最終的に衝突はしないが計画航路から大きく外れた針路を取ることになるといった問題が生じる場合がある．また，船舶の避航操船においては船舶の運動特性などを主たる理由として，最接近点（CPA）解析に代表されるように，現時点の各船舶の位置だけでなく船速や針路を考慮した将来的な衝突のおそれを考慮する必要がある．そのような複数の船舶の衝突のおそれを評価した際に，どのように複数の船に対する評価を統合して避航判断に用いるかという問題が生じる．

本研究では，広範囲の複数船舶を考慮した避航操船とウェイポイント航行を並行して行う深層強化学習モデルの構築を目的としている．衝突のおそれを判定するために，船同士が将来的に衝突する領域を表示する OZT を用いた．OZT の計算では，見合い関係を正確に表現するために船首航過距離（BCR）を考慮した計算法を示す．また，相手船との距離が設定された安全航過距離より小さい場合に OZT を補完することで深層強化学習を安定させる内部 OZT の計算法についても報告する．また得られた OZT を効率よく処理し，自船周りの OZT の分布を各成分が 2 値であるような 1 つのベクトルで表現する手法を開発し，これをグリッドセンサーと名付けた．避航操船の学習には，Actor-critic 法に分類される深層強化学習のアルゴリズムの一つである Proximal Policy Optimization

アルゴリズム (PPO)[2] を採用した．Actor と Critic の持つネットワークを離散的な行動空間において畳み込み層と全結合層を組み合わせたネットワークと，連続行動空間でさらに時系列データを取り扱うことのできる long short-term memory (LSTM)を組み合わせたネットワークの 2 つのネットワークを検証した．学習に用いる環境は近年の深層強化学習研究で標準的に用いられる環境フレームワークの一つである OpenAI Gym[10]に準処し実装した．この環境は KT モデルに基づいた船舶の運動計算などを含むシミュレーション機能や複数の方法によるシナリオの読み込みに対応しており，柔軟なシミュレーション環境を提供する．学習においては本環境を用いた数値計算によるシミュレーションを実施した．最後に学習モデルの性能に関して今津問題を対象とした数値シミュレーションによる検証を行った．また，5 隻の船舶が航行する複雑なシナリオを用いて，連続行動空間モデルの避航操船性能を検証した．

2.　Obstacle Zone by Target (OZT)

本章では，自船の周りを航行する相手船との衝突のおそれを評価する方法として，OZT の計算方法を示すとともに，前方航過距離を考慮した OZT の計算法および安全効果距離以下で計算される内部 OZT による OZT の補完方法について示す．

2.1 OZT の計算法

船舶の避航操船においては，避航のタイミングは自船と周囲を航行する相手船の動的情報をもとに最接近点までの距離（DCPA）と到達時間（CPA）によって判断される．別の方法としては，自船から相手船までの距離と自船からみた相手船の方位変化に基づいて避航操船の判断を行うこともある．いずれの方法においても，衝突のおそれを評価をできるのはあくまで現時点の自船の針路と船速に対してのみであり，自船が変針した場合に周囲の船舶との衝突危険がどのように変化するかは実際に変針してからでないと分からない．このような点を解決する方法として，本研究では自船と周囲を航行する相手船との間の衝突のおそれを，OZT を用いて評価する．OZT は，現在の位置，速力と針路でお互いの船が進むときに自船と将来的に衝突する可能性のある領域として表現される（Fig. 1 の青い線で囲まれた領域）．将来的に相手船と自船が衝突するおそれのある自船の衝突針路C_0は(1)式により求められる．

$$c_O = \begin{cases} Az \pm \alpha - \arcsin\left\{\frac{V_T}{V_O}\sin(Az \pm \alpha - C_T)\right\} \\ Az \pm \alpha - \pi + \arcsin\left\{\frac{V_T}{V_O}\sin(Az \pm \alpha - C_T)\right\} \ (V_T > V_O) \end{cases} \tag{1}$$

ここで$\alpha = \sin^{-1} r/d$でrは安全航過距離，dは自船から相手船までの距離．V_0およびV_Tはそれぞれ自船と相手船

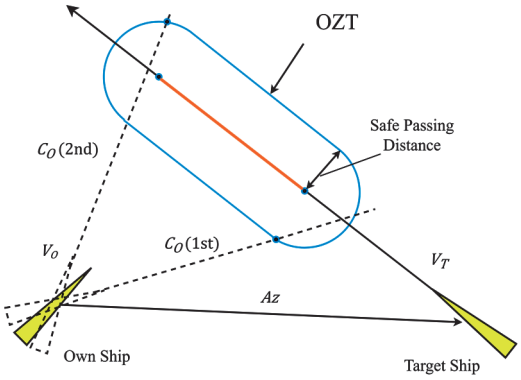

Fig. 1 Computation of OZT

の船速である．Azは自船から見た相手船位置の方位角であり，C_Tは相手船の針路である．これにより求められた最大 4 個の衝突針路C_0の組を用いて，それぞれの衝突針路を取った場合の DCPA および TCPA はC_0を用いて(2)式および(3)式により求められる．

$$\mathrm{DCPA} = d|\sin(C_R - Az + \pi)| \tag{2}$$

$$\mathrm{TCPA} = \frac{d\cos(C_R - Az + \pi)}{V_R} \tag{3}$$

ここでC_RおよびV_Rはそれぞれ自船から見た相手船の相対針路と相対速度である.この TCPA をもとに相手船の針路上に Fig.1 で示される，線分から半径が安全航過距離となる円スイープ形の領域が描け，これが OZT となる．

2.2 前方航過距離を考慮した OZT の計算

通常の OZT の計算では，前節で説明したとおり，相手船周りに半径が安全効果距離となるような円領域が自船と重なるような領域として表される．一方で，船舶が行き会う場合には船舶領域（Ship Domain）や閉塞領域（Effective Domain）といった指標に代表されるように基本的に船舶の前方はその他の方向と比べて航過距離を大きくとることが知られている．OZT の計算において相手船の前方航過距離を考慮する際は，対象となる相手船の前方に，確保したい前方航過距離から安全航過距離を引いただけの距離を十分に埋めるように先を行く同速の仮想船を考え，この仮想船を用いて計算した衝突針路の TCPA を使って OZT を計算し，その分，本来の相手船に対して計算された OZT を延長する方法がある．また他の単純な方法としては確保したい前方航過距離から安全航過距離を引いた分だけ元の OZT を延長することで対応することもできる．後者の方法では，衝突針路が変わるために単純に前方航過距離分だけ延長したものは，前者の方法と厳密には差が出るが，本研究ではその差は十分小さいとして後者の近似的な方法を採用した．このように計算された前方航過距離を考慮した OZT と従来の OZT について Fig. 2 に示す．左右舷側からの横切りのシチュエーションにおいて，前方航過距離がない場合は，OZT のみでは区別がつかないが，前方航過距離を考慮した OZT では区別することができる．また後述の深層強化学習を用いた学習を行う上で，相手船の前方航過距離を考慮した OZT に対応した衝突判定をおこなうために，自船が相手船の周りの Fig. 3 に示すような領域に侵入したときに衝突と判定する．この領域は安全航過距離の半径を持つカプセル型の領域で，船体中央から船首方向に設定した前方航過距離と等しい長さを持つような形状である．

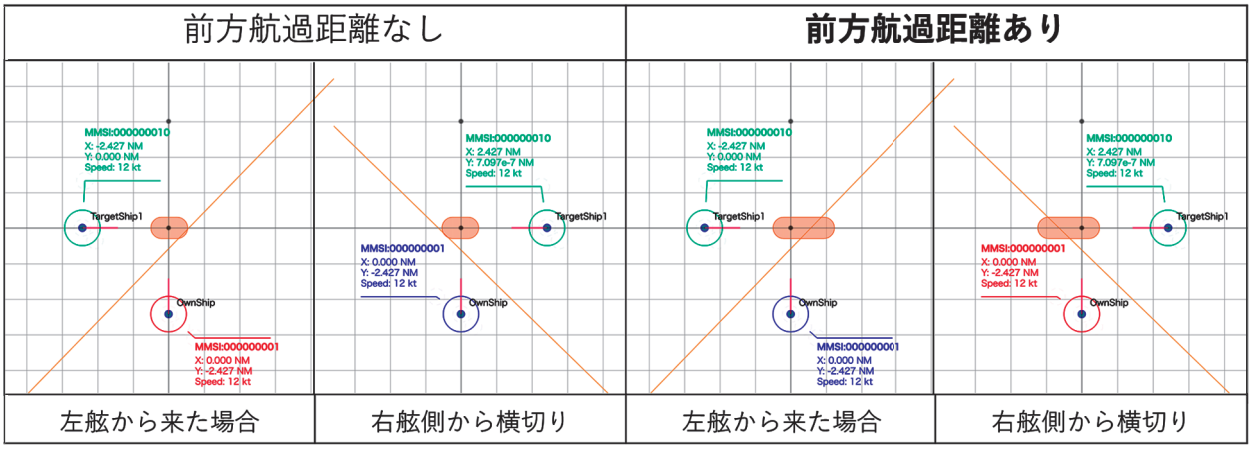

Fig. 2 OZT with the bow crossing range in crossing encounter situations

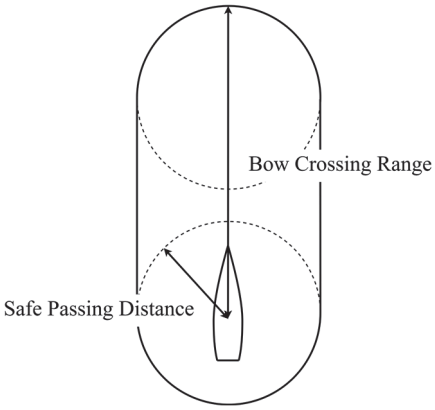

Fig. 3 Domain for collision detection

2.3 内部 OZT

OZT は(1)式を改めてみると，arcsinの計算があるために 2 船の距離が安全航過距離より小さい場合に計算することができない．そのため，輻輳海域などでやむを得ず安全航過距離以下の距離まで接近した場合には相手船の OZT が計算できなくなるため，OZT のみを頼りにした避航判断を行う場合には危険である．これを回避するために相手船との距離が設定した安全航過距離以下になった場合に OZT に類する指標を提示することが必要である．これを内部 OZT と呼んで通常の OZT と区別する．内部 OZT は自船からみて相手船との距離が安全航過距離以下，つまり$d < r$となったときに，相手船針路上にTCPA $= 0 \rightarrow r/|V_O - V_T|$の間に相手が進む区間を中心とした半径$r$となるような領域として定義される．Fig. 4 では，自船の前を 2 隻の船が横切っているときの OZT を示している．Target Ship 2 は自船との距離が安全航過距離より大きいため通常の OZT として表示されている．一方で自船のすぐ前方を東へ横切る Target ship 1 は緑の円で示される安全航過距離の内側に自船が来てしまっているため，内部 OZT によってその衝突予想範囲が示されている．次章で述べる深層強化学習を用いた学習を行うシミュレーションの過程では相手船の存在は OZT のみにより検知できるので，相手船の OZT が表示されていない場合に「周りに衝突のおそれのある相手船が存在しない場合」と「ある船との距離が安全航過距離以下になったために OZT が表示されていない場合」の区別をすることができない．このため学習が進んでも状況の良し悪しを評価する価値関数のネットワークの損失関数の値が下がらない現象が起きていた．これはつまり学習がうまく進んでいないことを表している．内部 OZT の導入により衝突時に得られる報酬と OZT 検知結果を含む状態ベクトルの関連が保てるようになるため，Fig. 5 に示すように価値関数の損失関数の値が安定して減少するようになり学習がより安定することが確認できた．

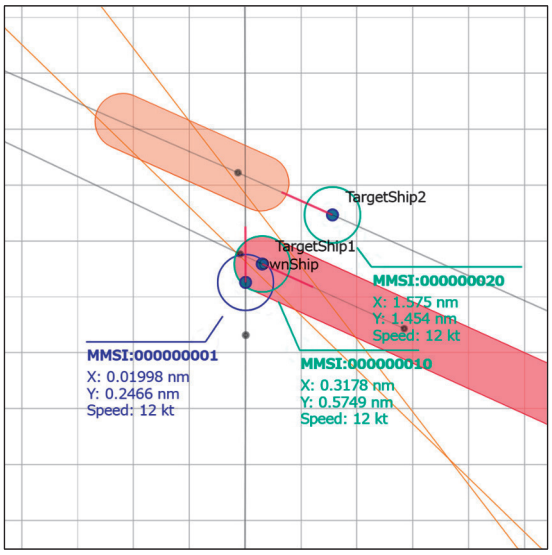

Fig. 4 Inside OZT (red area)

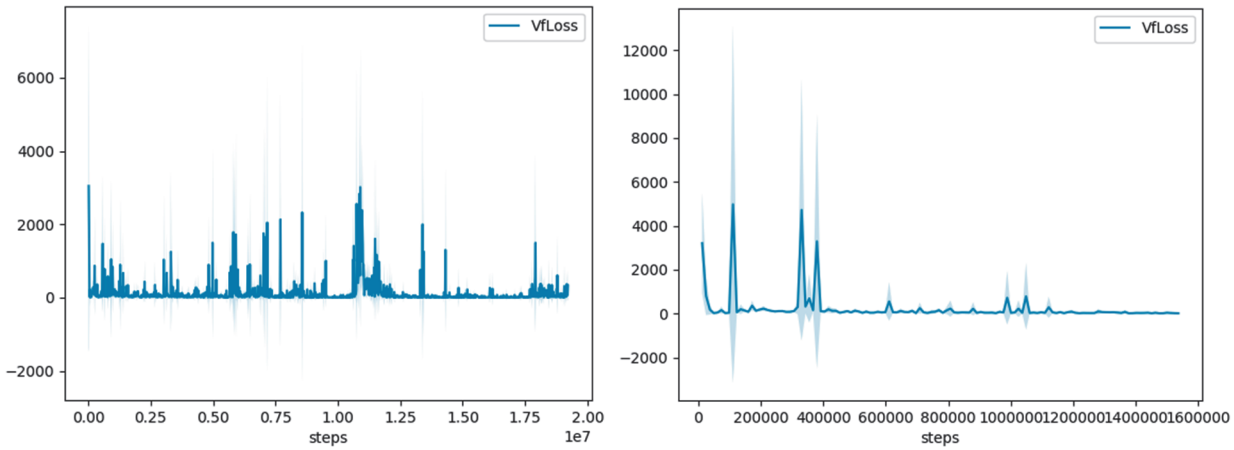

Fig. 5 Comparison of changes of loss function for value network with/without inside OZT (left: not using inside OZT, right: using inside OZT)

2.4 OZT の検知手法：グリッドセンサー

2.1 で述べた OZT は，将来的に衝突の可能性のある領域を表示する方法である．OZT は非常に有用であるが，一方でこのままでは次章で述べる深層強化学習へは適用しづらいため，入力として利用しやすい形へ変換する必要がある．なぜなら，強化学習はじめとする多くの制御理論においては事前に入力ベクトルの次元数が決定されている必要があり，学習から検証の全過程においてその次元数を変えることはできないためである．また，OZT の数は，相手船の数や自船との見合い関係によってその数が変わるため，OZT の数によらず固定次元のベクトル表現を得られると都合が良い．

一つの方法としては，自船から何本かの仮想の検知線を放射状に伸ばし，この検知線と相手船の OZT との重なり位置を入力とする方法である．この手法は 2 つの問題を抱えており，一つは，検知線を放射線上に伸ばすと，遠くになるほど検知線の間隔が広がり検知漏れを引き起こす可能性が高くなる． もう一つは，通常一本の検知線は同時に一つの対象までしか検知できないという制限があるため，例えば一つの検知線に複数の対象が重なった場合には最も近いもののみ検知されそれより遠方のものは無視されてしまう．これでは複数の船舶を検知可能とは必ずしも言えない. 船舶はそれ自体の操縦運動特性のために周囲の船舶の動きを検知することが求められるが，放射状の検知線による方法では広い範囲の船舶を考慮するためには課題が残る．そこで，これらの問題を解決できる手法として Fig. 6 に示すように自船の周囲を同心円状のグリッドで分割した仮想センサーであるグリッドセンサーを設計した.

グリッドセンサーは角度方向と動径方向を等分割された同心円グリッドで構成されており，センサー上の分割されたそれぞれのセル毎に OZT との重なりを判定することで OZT の検知を行う．OZT との重なりを 0-1 で表現し，全体としてセルと同数の成分を持つ一つのベクトルとすることで OZT の分布を表現することができる．Fig. 6 では薄い緑色の線で区切られたものがグリッドセンサーであり，図中央下部，グリッドセンサーの中央に位置する黄色い船は自船，自船の左舷側から来る船がここでは相手船である．図中左部から進む相手船の針路上に示された青い曲線で囲まれた領域が OZT である．図中の赤いセルはグリッドセンサーの検知により OZT と重なったセルである．このようにして自動避航操船アルゴリズムは，OZT の情報を一つの固定次元ベクトルとして認識する．また本報告では行わなかったが，同様の方法で陸地や水深の関係で航行が制限される区域やブイなどの障害物のような類の操船判断に影響を及ぼす要素も，二次元の図形として専有領域を表現できるものはグリッドセンサーを通して統一的に処理することができる．この方法であれば密に配置した検知線による入力と異なり，画像のように 2 次元の配列に整形することで，畳込み層などを持つニューラルネットを通して 2 次元的な特徴量を抽出することも可能であり，より本質的な衝突危険箇所の表現を得ることができる.

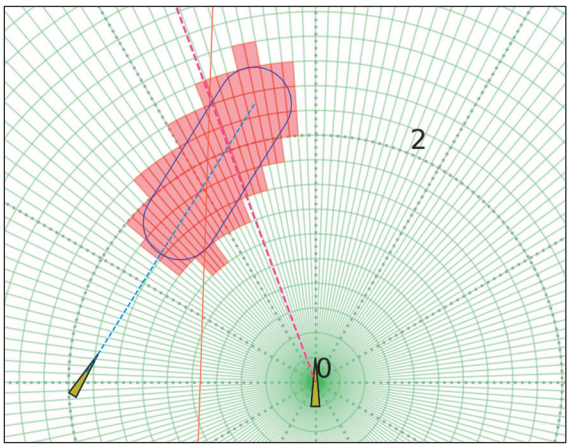

Fig. 6 Detection of OZT by the grid sensor (The unit of radius is nautical mile and size of ships' plots is 4 times of full-scale)

3.　深層強化学習による避航操船の学習

本章では，OZT に基づいた複数船舶の運航状況を考慮した避航操船を，深層強化学習を用いて学習する方法について述べる．

3.1 深層強化学習

避航操船を習得した学習モデルを構築するために，本研究では深層強化学習と呼ばれる手法を用いる．深層強化学習は，その名の通り強化学習と深層学習を組み合わせた機械学習の一手法である．強化学習の「強化」は元々行動心理学において動物が報酬や罰によって行動選択に対する条件づけを指す．例として，ネズミを押すと餌が落ちるレバーの付いた箱の中にいれると，餌をもらうためにレバーを押すことを学習するような現象に着想を得たのが強化学習である．理論的には直前の状態のみに基づいて次の状態が決定されるマルコフ性を前提として定式化される．強化学習では，与えられた環境においてエージェントが各タイムステップ$t = 0,1\cdots$における状態$s_t \in S$（Sは環境が取りうる状態の集合）を観測し，これに基づいて行動$a_t \in A(s_t)$（$A(s_t)$は状態s_tでエージェントが取ることのできる行動の集合）を選択する．行動を取って状態が遷移するたびにエージェントは予め定義された関数$R(s_t, a_t)$に基づいて報酬r_tを獲得し，これを繰り返して最終的に得られる報酬の和を最大化するような方策πを学習する．エージェントは行動を取ったあとに環境から得られる報酬と次のタイムステップの環境の状態を観測して次の行動を取ることを繰り返し，最終的に累積報酬を最大化するように学習を行う．よって行動の良し悪しは報酬によって特徴づけられる．避航操船の問題においては，エージェントは自船に対応し環境は相手船舶やウェイポイントなどの避航操船の判断に影響がある諸要素より構成される．

環境の状態を基に次に取る行動を決定する方法にはいくつか種類がありまたモデルの更新方法などに応じて，強化学習には様々なアルゴリズムが提案されている．強化学習では，基本的には状態を引数とする関数（これを方策と呼ぶ）の出力をもとに行動を決定する．たとえば Actor-critic 法と呼ばれるアルゴリズムでは，オンポリシー型の場合，学習された方策が表す行動の確率分布を基に行動を選択する．また，各状態に対する将来的に得られる報酬の期待値で表現される「価値」として表し，状態と価値を対応付ける価値関数（Critic）を更新する．この価値関数から推定される状態の価値に基づいて行動の良し悪しを評価することで方策を更新する．この方策や価値関数を，パラメータを持つ関数で表現することを関数近似と呼び，深層強化学習では（ディープな）ニューラルネットを用いる．ニューラルネットによる表現力を利用することで，画像のような高次元の入力にも対応したエージェントの複雑な制御を可能にしたのが深層強化学習である．

3.1.1 深層強化学習アルゴリズムの実装

深層強化学習は，状態の評価や行動の決定方法，価値関数や方策などの更新手順等により，様々な種類のアルゴリズムが存在する．本論文では GPU を用いた高速な計算に対応した深層学習フレームワークである Pytorch[9]により実装された深層強化学習ライブラリ machina[7]を用いて実装を行った．実際の学習では machina で実装された Proximal Policy Optimization (PPO)アルゴリズムを採用した．以下は，machina の実装にける PPO アルゴリズム

の説明が含まれる．PPO は，更新前後の確率分布の Kullback-Leibler ダイバージェンスに応じて方策更新に制限をかける Trust Region Policy Optimization (TRPO)[3]と，分散学習と Advantage という値を使って更新を行う Asynchronous Advantage Actor-Critic (A3C)[4]の 2 つの Actor-Critic 法の強化学習に源流を持つアルゴリズムである．Actor-critic 法では，基本的に 2 種類のネットワークを用意し学習を行う．一つは観測した状態に対する事後確率分布の形で行動を決定する方策$\pi(a_t|s_t;\theta)$と，もう一つは状態を評価する価値関数$V(s_t;\theta_v)$である．ここで下付きのtはタイムステップを表し各タイムステップにおける行動をa_t，環境の状態をs_tとし，θとθ_vはそれぞれ方策と価値関数を表現するネットワークのパラメータである．本研究の実装では方策と価値関数の 2 つのネットワークは共有部分を持たず，それぞれ独立に更新される．方策は，ある状態の下で取った行動の価値を表す Advantage に基づいて更新される．Advantage の推定値である$A(s_t,a_t;\theta,\theta_v)=\sum_{i=0}^{k-1}\gamma^i r_{t+i}+\gamma^k V(s_{t+k};\theta_v)-V(s_t;\theta_v)$を用いて，$\nabla_\theta \log\pi_\theta(a_t|s_t;\theta)A(s_t,a_t;\theta,\theta_v)$のように計算される勾配を用いて更新が行われる．ここでγは割引率，rは報酬である．machina の実装においては，この Advantage の推定値をバッチ内の Advantage の推定値の分散と平均値を用いて正規化を行っている． PPO では方策の更新の際に TRPO に倣って更新目標となる分布を修正して学習を行う点が最大の特徴となっている．PPO には主に 2 種類の実装があるが，このうち Clip と呼ばれる種類のアルゴリズムでは(4)式に示すように目的関数を設定し，値の更新に制限をかけることで方策の更新の安定化を図っている．

$$L^{CILP}(\theta)=\hat{E}_t\left[min\left(r_t(\theta)\hat{A}_t,\ clip(r_t(\theta),1-\varepsilon,1+\varepsilon)\hat{A}_t\right)\right] \tag{4}$$

ここで$r_t(\theta)=\pi(a_t|s_t;\theta)/\pi(a_t|s_t;\theta_{old})$で$\varepsilon$はハイパーパラメータ，$\hat{A}_t$はタイムステップ$t$に推定された Advantage を表す. 価値関数については Monte Carlo 法を用いており, 具体的には割引報酬和$G_t^T=r_{t+1}+\gamma r_{t+2}+\cdots+\gamma^{T-1}r_T$と現在の価値関数の値$V(s_t;\theta_v)$との差の 2 乗を損失関数として, これを最小化するように更新を行う．

PPO に代表される Actor-Critic 型の強化学習アルゴリズムの特徴は行動の決定を行う方策関数と，現在の状態を評価する価値関数を分けて学習を行う点にあり，特に価値関数の出力層を変更するだけで，離散行動空間と連続行動空間の両方において学習を行うことができる．

3. 1. 2 ネットワークの設計と更新

本研究では，2 種類の行動空間に対応したネットワークを用いて学習を行った．これを Fig. 7 に示す．環境の状態として 1.グリッドセンサーによる OZT の検知結果, 2.自船の方位角, 回頭角速度, 船速, 舵角正規化した値, 3.ウェイポイントまでの方位角，距離およびウェイポイントをしたときのオートパイロットの司令方位を正規化した値の情報を深層強化学習の方策と価値関数を表現するネットワークに入力する．またオートパイロットで制御を行ったモデルでは，ウェイポイントに関する 3.の情報の内，オートパイロットから計算された司令舵角の代わりに自身の現在のオートパイロットの司令方位を正規化した値を入力した．各ネットワークに共通する特徴としては，グリッドセンサーで検知された OZT の処理と，自船の動的情報やウェイポイントなどに関する情報の処理を分けて入力している点である．グリッドセンサーの検知結果は 1 つのベクトルとして表現されているが，実際には 2 次元のテンソルとして解釈した方が実情をより表現できることは明らかである．そこでグリッドセンサーからのデータは畳み込み層を通し，その他の情報は基本的な全結合層により処理した上で両者を統合し，最終的な出力を得る構造を取っている．

本研究でいくつかの設定において学習を行った際，最初に離散行動空間で学習を行ったが，安全航過距離を設定に限界があることがわかった．理由として考えられるのは今回使用した machina の実装では離散行動空間の学習で時系列データを扱うことのできる再帰ニューラルネットワークの層を利用できなかったために性能に限界があったと考えた．そこで今回は，連続行動空間で RNN の一種である Long Short Term Memory (LSTM)と呼ばれるセル(LSTM は層ではなくセルと呼ぶ場合が多い)をネットワークの出力層の手前に挟んだネットワークを設計した（Fig. 7 の上 2 つのネットワーク）．連続行動空間のネットワークに関しては，舵角制御とオートパイロット（目標針路の出力）の 2 種類のネットワークを用意した．一方離散行動空間のネットワークでは，舵角を出力する舵角制御の 1 種類のネットワークで学習を行った．ここで舵角制御のモデルのみ方策のネットワークでは畳み込み層を 2 層にし，価値関数は畳み込み層を 1 層だけ持つ．ネットワークの更新には Adam を用いた．

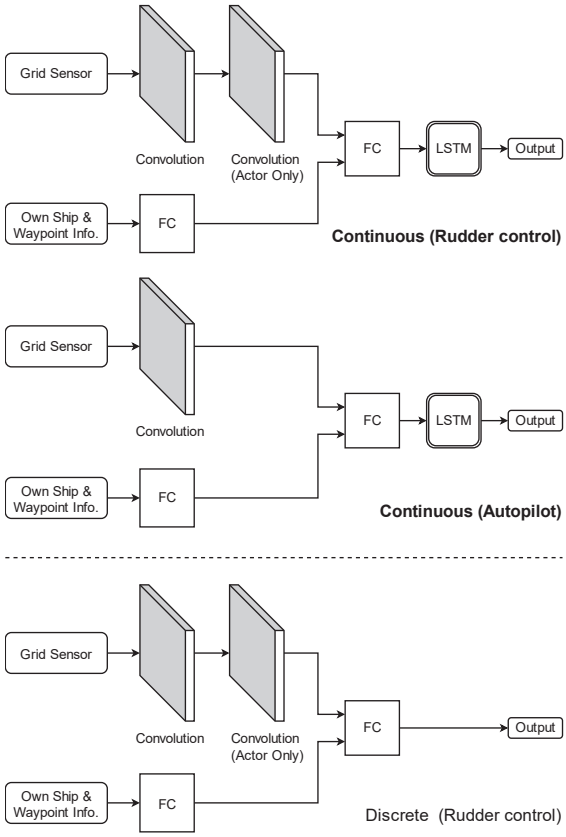

Fig. 7 Illustration of a network in PPO

3.2 学習のための環境設計

避航操船を学習するための（強化学習における）環境は，船舶，ウェイポイント，対象海域（ゲーミングエリア）などによって構成される．離散行動空間と連続行動空間の学習においてはほとんどの設定が共通であるが一部が異なっている．設定の詳細は文献[11),12)]を参照されたい．学習に用いるシナリオとしてはここでは今津が用いた，見合い関係を集めた問題群で今津問題[6)]を題材として選んだ．今津問題に用意されている見合い関係を Fig. 8 に示す．Fig. 8 中において三角のマークで表されるのが自船，丸で表されるのが相手船であり，それぞれの針路を線分の向きで表している．枠内左上の数字が問題のケース番号を示し，今回用いる今津問題は 22 ケースの問題が用意されており，どれも枠内の中央の地点に同時に到着するように，つまり衝突するように配置されている．ここで Cai と長谷川による研究[5)]から，少なくともこの問題では相手船による避航操船を許すと設定した衝突危険が解消されてしまうことにより問題の難易度が易化するもしくは想定したシナリオが大きく変化してしまう可能性があることから，相手船はウェイポイント航行や避航操船等の変針を一切行わず直進するように設定した．そのため，今回は相手船同士の衝突は無視する．加えて，学習モデルの汎化性能を上げるために 3 船がランダムに配置される特殊なケースを一つ用意し，今津問題の 22 ケースと合わせて計 23 ケースの問題のもとで学習を行った．今津問題における各ケースの初期位置と初期方位角は文献[11),12)]を参照されたい．今回は設定速力のときに衝突までの時間が 30 分になるように船舶の位置を決定した. 自船は(0, -6)[NM]の地点に配置され, 方位角を-5~+5deg. の範囲でランダムに設定した．シミュレーション上の学習および検証においては，自船，相手船ともに船速 12 ノットに設定し， 各ケースにおける全船舶が 30 分後に原点に到着するように配置した．ただし，ケース 3 などに含まれる被追い越しを想定した針路が 0.0 deg.の相手船の船速は 70%にあたる 8.4 ノットに設定した．

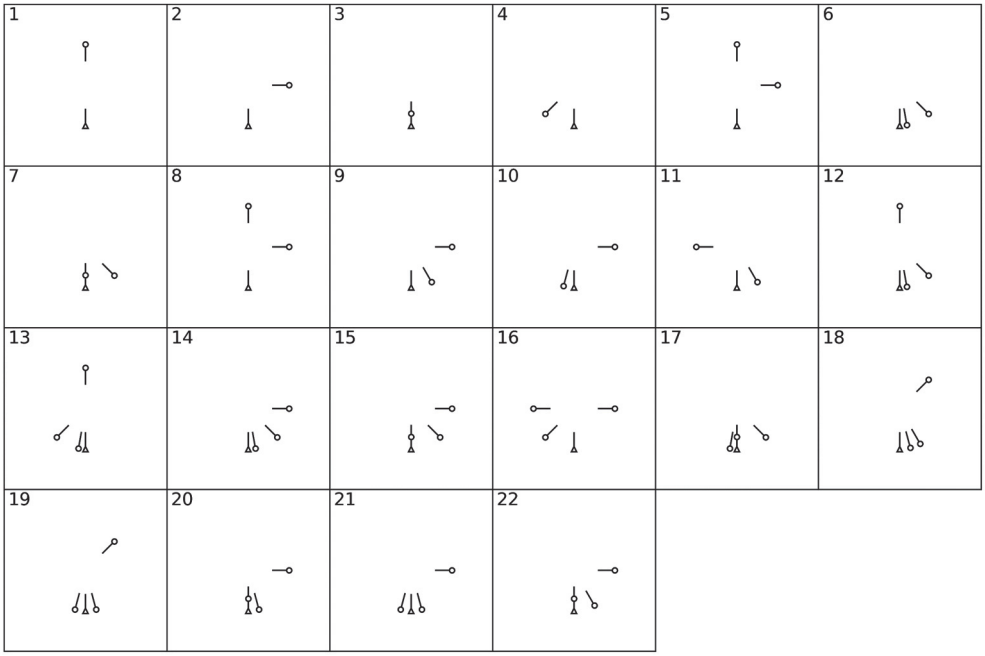

Fig. 8 Imazu problems

船舶の操縦運動計算は野本の一次遅れの応答モデル（KT モデル）[14]，舵角の変化は司令舵角による一次遅れの形で表され，これらを状態方程式の形で表したのが(5)式である．

$$\begin{bmatrix}\dot{\psi}\\ \dot{r}\\ \dot{\delta}\end{bmatrix}=\begin{bmatrix}0 & 1 & 0\\ 0 & -1/T & K/T\\ 0 & 0 & -1/T_E\end{bmatrix}\begin{bmatrix}\psi\\ r\\ \delta\end{bmatrix}+\begin{bmatrix}0 & 0 & 0\\ 0 & 0 & 0\\ 0 & 0 & \delta_C/T_E\end{bmatrix} \tag{5}$$

ここでψは方位角，rは回頭角速度，δは舵角でδ_Cは司令舵角を表す．TとT_Eはそれぞれ方位角と舵角の変化に対する時定数であり，Kは操舵量に対するゲインである．3.1.2 で述べたように，今回は司令舵角を出力する舵角制御のモデルと司令方位の変化分を出力するオートパイロットの 2 種類の制御の方法で学習を行った．今回の計算に使用した船舶の要目を Table 1 に示す．これらは自船・相手船を問わず，シミュレーション上のすべての船舶で共通である．舵角制御のモデルでは司令舵角は，-20～+20 deg.の範囲から現在の方策を基に決定する．オートパイロットで制御するモデルでは現在の方策に従って-10～+10 deg.の範囲の値を選び，各タイムステップでオートパイロットの現在の司令方位に加算する．

提案手法では，基本的に位置情報を直接扱わないことで問題依存のモデルの学習を回避するように努めた．本アルゴリズムでは相手船の動的情報は OZT の検知結果のみから把握される．グリッドセンサーの検知および方策による司令舵角の更新はシミュレーション内の時間で，離散行動空間では 5s，連続行動空間では 10s おきに行い，運動計算については積分計算の時間間隔を 1s とした．そのほかの学習時の環境の設定については Table 2 に示す．本研究では，強化学習向けの開発・評価用プラットフォームで標準的に利用されている OpenAI Gym のインターフェースを利用し，Python により環境を実装した．これにより，今回用いたものとは別の深層強化学習アルゴリズムを試す際にも本環境に容易に適用する事ができる．

Table 1 Subjects of ships for learning

K [1/s]	0.05
T [s]	50.0
T_E [s]	2.5
L_{PP}[m]	106
B [m]	16.2
U [kt]	12.0, 8.4 (ships over-taken only)

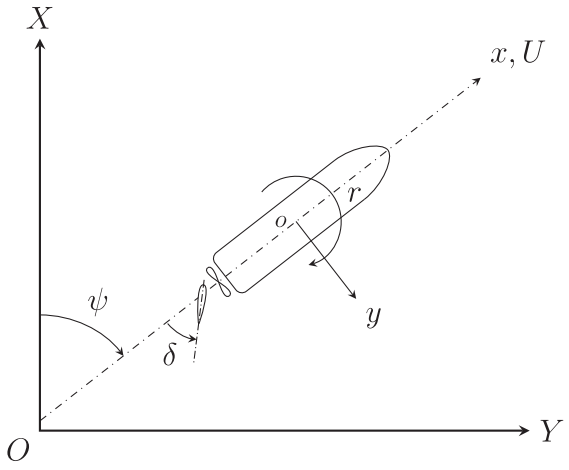

Fig. 9 Coordinate system of ship motion

Table 2 Configurations of environment

Subjects	Value
safe passing distance [NM]	0.5
Grid Sensor	
angle of detection [deg.]	360.0
grid spacing on angular direction [deg.]	2.0
radius of sensor [NM]	12.0
grid spacing on radius direction [NM]	0.2
detection intervals [sec.]	10.0

3.2.1 報酬の設計

報酬については，各ステップで逐次加点される基礎報酬とエピソードの終了時に与えられる成果報酬に区別し設計を行った．ここでエピソードとはシミュレーションのスタートから次の条件を満たしシミュレーションを打ち切るまでを指す．エピソードの終了条件は，ウェイポイントとの距離が規定の値以下になるか設定したステップ数に達するかのいずれかを満たしたときとした．基礎点$Costs$は， 離散行動空間の学習では(6d-9d)式，連続行動空間の学習では(6c-9c)式，のように定めた．COLREGs を意識した操船を行うために， $Costs_{left}$として小さな正の報酬を与えることで，自船は右舷側に避航するように学習を行う．

（離散行動空間）

$$\mathrm{Costs} = \mathrm{Costs_{wp}} + \mathrm{Costs_{left}} \tag{6d}$$

$$\mathrm{Costs_{wp}} = (0.01\mathrm{Costs_{Az_{wp}}}\tanh(1/\mathrm{d_{wp}}) \tag{7d}$$

$$\mathrm{Costs_{left}} = \begin{cases} 0.05, & \mathrm{Az_{wp}} \geq 0 \\ 0.0, & \mathrm{Az_{wp}} < 0 \end{cases} \tag{8d}$$

$$\mathrm{Costs_{Az_{wp}}} = (\pi/4 - \min(|\mathrm{Az}|, \pi/4))/(\pi/4) \tag{7d}$$

（連続行動空間）

$$Costs = Costs_{wp} + Costs_{left} + Costs_{stable} \tag{6c}$$

$$Costs_{wp} = 0.9\,tanh(1/d_{wp}) \tag{7c}$$

$$Costs_{left} = \begin{cases} 0.05, & Az_{wp} \geq 0 \\ 0.0, & Az_{wp} < 0 \end{cases} \tag{8c}$$

$$Costs_{stable} = -0.01|r/\pi| \tag{9c}$$

ここで, d_{wp} と Az_{wp} はそれぞれ自船から見たウェイポイントの距離と方位角である. 成果点としては衝突なくウェイポイントから規定の距離以内に到達したときは＋50 と設定した. また衝突の判定がされたステップで連続行動空間モデルではエピソードを打ち切らずに-5 の報酬を追加で与えた. 一方離散行動空間の学習では, エピソードを打ち切り-50 の成果点を与えた.

4. 学習モデルの検証

4.1 今津問題による検証

今回検証に使用したのは離散行動空間の避航モデルと 2 種類の連続行動空間の避航操船モデルの計 3 種類のモデルである. これらのモデルを用いて, 今津問題の全 22 ケースを使ってシミュレーションを行い, 避航性能の検証を行った. 今津問題, 全 22 ケースにおける航跡の例を Fig. 10, 11, 12 に示す. また, 各ケースにおける最小離隔距離を Fig. 13 に示す. 離散行動空間のモデルが最も針路が安定しているが, 一方で安全航過距離を 0.5NM にすると学習することが出来なかった [11),12)]. 0.5 NM という距離は, 300m 程度の船を想定した際のバンパーモデルにおける左右舷方向の長さに対応する [11)]. 連続行動空間のモデルにおいては, オートパイロット制御のモデルでは, 変針が大きくなる傾向がみられる. 航過距離が大きくなると, ウェイポイントまでの到着時間が伸びるため好ましくない. その点相対的に連続行動空間の舵角制御のモデルでは 0.5NM の安全航過距離を確保しつつ, 複雑な見合い関係でも原針路からの逸脱を抑えることができている.

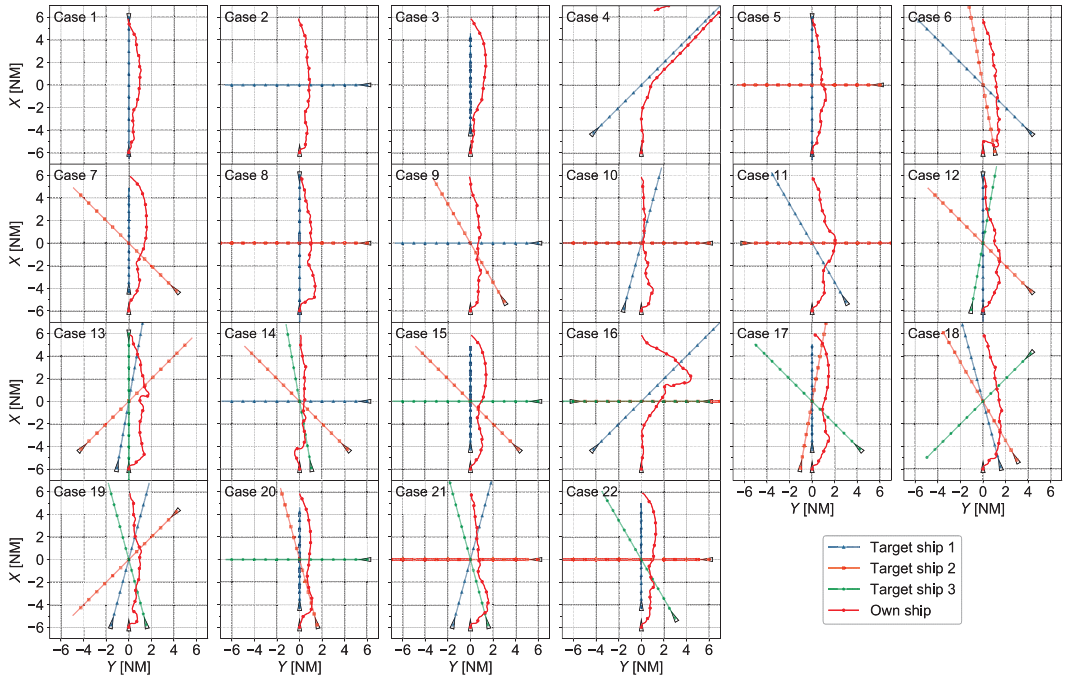

Fig. 10 Trajectries through Imazu problems using the continuous action space model

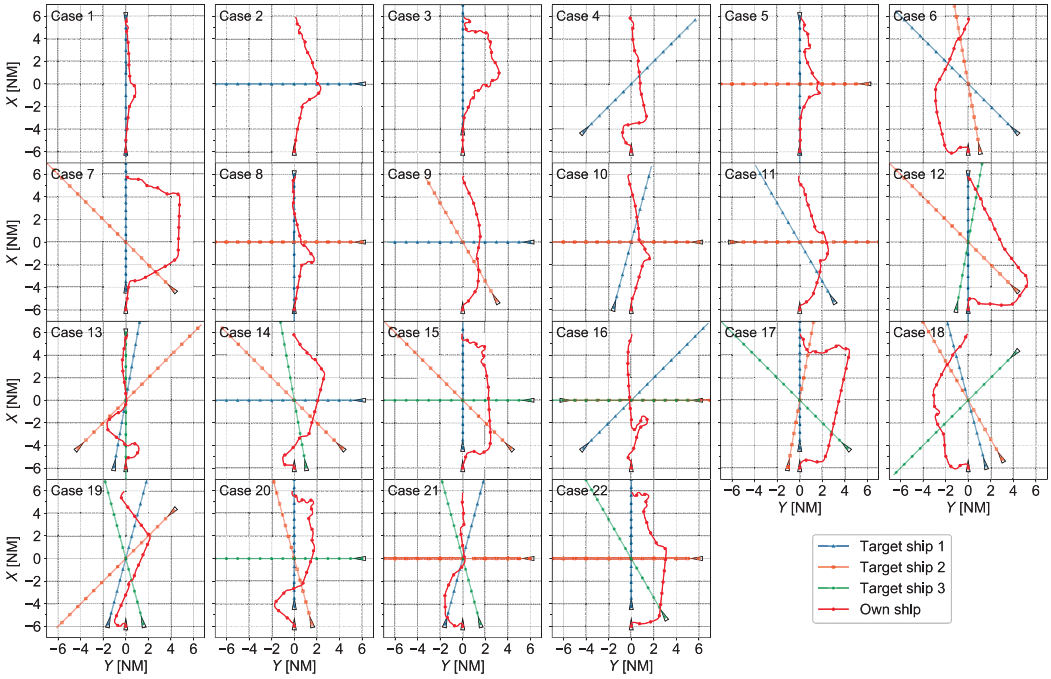

Fig. 11 Trajectries through Imazu problems using the continuous action space model with autopilot

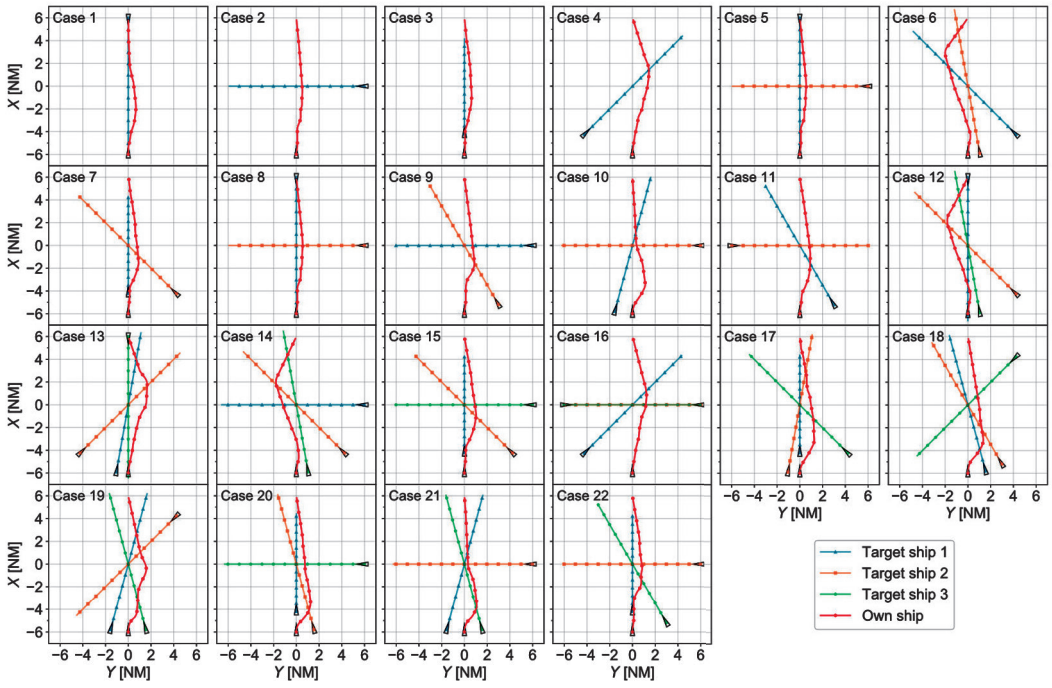

Fig. 12 Trajectries through Imazu problems using the discrete action space model (the safe passing discance is 0.3NM)

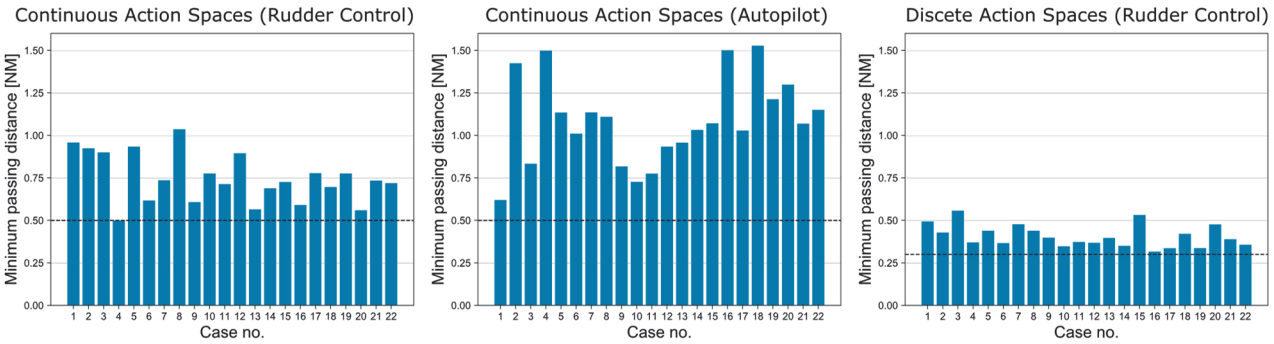

Fig. 13 Minimum passing distance of trained models (the safe passing distance for the trained models in continuous action spaces is 0.5 NM, and it for the model in discrete action spaces is 0.3 NM)

ここで，舵角制御のモデルは，Fig. 14 に示すような旋回をして目的地に向かうといった高度な制御を会得している．こうした大胆な変針を伴う避航操船は他のモデルでは一切見受けられなかった．このような操船が学習されたのは，学習時に経過時間に対する負の報酬設定を課していないことが起因していると考えられる．とはいえ，実際の操船では経済性も考慮する必要があるためこうした柔軟な制御を残しつつ，時間的にも効率的なモデルを構築できるように報酬を設計する必要があると考えられる．

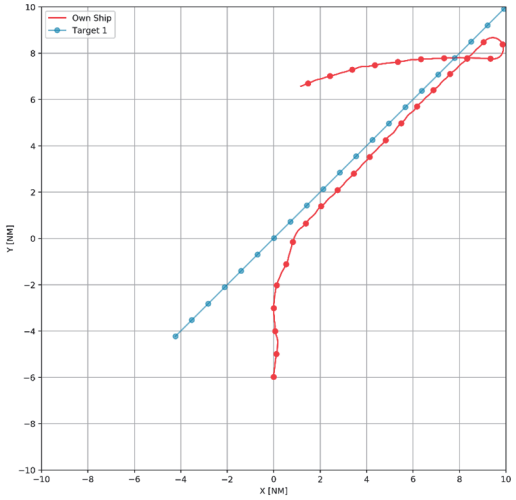

Fig. 14 Trajectory in case 4 of Imazu problem in Fig. 10

4.2 船舶 5 隻が航行するシナリオでの検証

最も性能が高い連続行動空間の舵角制御モデルを対象として，さらに学習モデルの汎化性能を検証した．検証のために，今津問題に含まれる最大隻数より多い 5 隻の相手船が航行するテストシナリオを準備した．テストシナリオの初期配置を Fig. 15 に示す．自船が目標とするウェイポイントは(0.0, 10.0)[NM]に設定されている．Fig. 13 に示されている OZT の配置から，自船がウェイポイントにたどり着くためには，OZT の間を抜けるように針路を取るか，もしくは大きく迂回する必要がある．

シミュレーション結果は Fig. 16 に示す．今回学習されたモデルは OZT の間を縫うように避航操船を行っており，テストシナリオに対して効率的な操船を取ることができていることがわかる．本テストシナリオにおける最小離隔距離は 0.753NM であり，設定した安全航過距離 0.5NM に対して余裕のある結果となった．結果的に学習に用いた今津問題だけでなく，今回試したテストシナリオにおいても，複雑な見合い関係においても適切な判断を行うことのできる学習モデルが構築できたことを確認した．

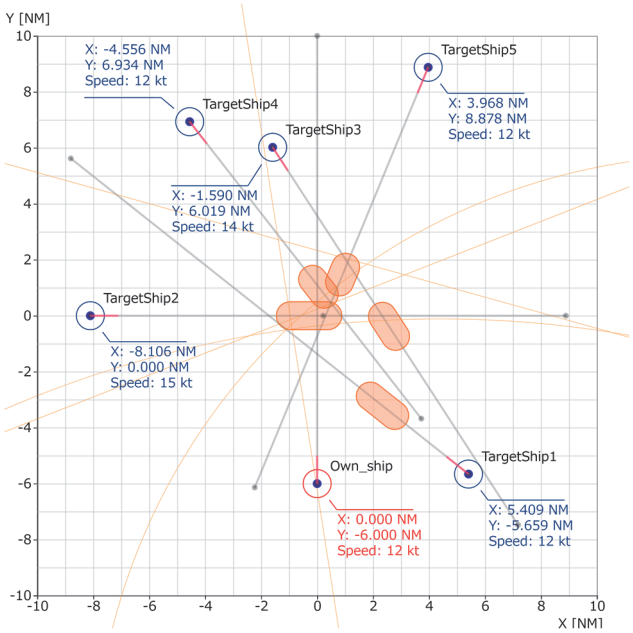

Fig. 15 Initial position and OZT distributions of the test scenario

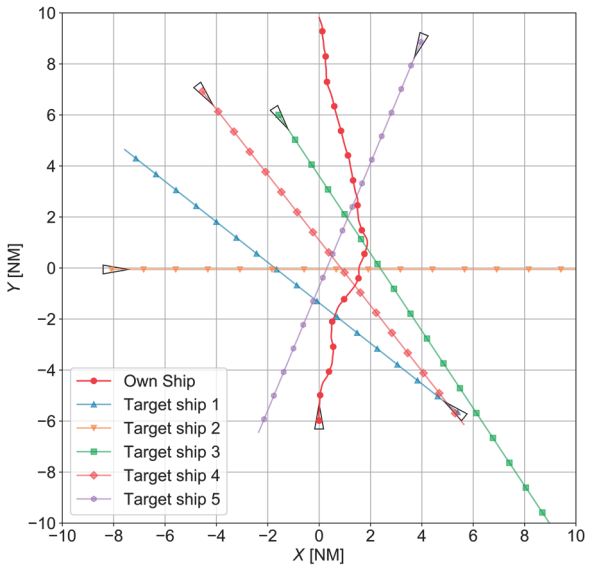

Fig. 16 Trajectory by the continuous action space model with rudder control for the test scenario

5. まとめ

本研究では，複数船舶の避航操船とウェイポイント航行を並行して行う手法としてグリッドセンサーによる OZT の検知と深層強化学習を組み合わせた手法を提案し，シミュレーションによる検証結果を報告した．相手船との衝突のおそれを評価するために OZT を採用し，さらに見合い関係の正確な把握のための前方航過距離の考慮と，学習の安定化のための内部 OZT という 2 つの手法を提案した．学習では今津問題を対象とし，離散行動空間と連続行動空間で 3 つの設定で学習を行うことで，特に前方航過距離と内部 OZT および LSTM を導入した連続行動空間では，より大きな安全航過距離においても避航操船を学習することができた．

連続行動空間での舵角制御の学習モデルでは，他のモデルでの学習では見られなかった途中で旋回を行うような複雑な制御を行うことのできる学習モデルを構築できることを確認した．また，同モデルに関しては学習時よりも隻数の多いシナリオにおいても，論文に示したシナリオにおいて，適切に避航操船を行えることを確認した．

今回は相手が変針を行わないシナリオを対象としたシミュレーションを行ったが，相手船の不意の変針に対する対応などより高度なモデルの構築手法を模索したい．

一方で，特に連続行動空間の学習モデルは針路の安定性が十分でなく，このままでは相手船に不安を与える操船になるおそれがあり今後の課題である．また，深層強化学習を用いたアルゴリズムをそのまま適用する上で，学習モデルの動作がブラックボックスであることが課題として指摘される．この点に関しては，学習モデルを様々なシナリオで動作させ学習モデルの判断を解析するなどしてホワイトボックス化することが実用上必要になると考えており，今後の課題としたい．

謝　辞

本研究の一部は，JSPS科研費19H02371，20K14971の助成を受けた．ここに記して関係各位に謝意を表する．

参考文献

1) J. M. Veras *et al.,* 2017, MAXCMAS project: Autonomous COLREGs compliant ship navigation, Proceedings of the 16th Conference on Computer Applications and Information Technology in the Maritime Industries (COMPIT)，pp.454-464.
2) J. Schulman *et al.,* 2017, Proximal Policy Optimization Algorithms，arXiv preprint arXiv:1707.06347.
3) J. Schulman et al., 2015, Trust region policy optimization. In International Conference on Machine Learning，pp. 1889-1897.
4) V. Mnih, et al., 2016, Asynchronous Methods for Deep Reinforcement Learning, Proceedings of the 33rd International Conference on Machine Learning，pp.1928-1937.
5) Y. Cai and K. Hasegawa, 2013, The Evolution of Marine Traffic Simulation System through Imazu Problem, Proc. JASNAOE, pp.191-194
6) 今津隼馬：避航法に関する研究，博士論文（1987），東京大学.
7) DeepX, Inc. : machina ~A library for real-world deep reinforcement learning ~, https://machina-rl.org/, 2019.
8) D.P. Kingma, J. Ba, 2014, Adam: A Method for Stochastic Optimization, Proceedings of the 3rd International Conference on Learning Representations.
9) A Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga and A. Lerer, 2017, Automatic Differentiation in PyTorch, NIPS Autodiff Workshop.
10) G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang and W. Zaremba, 2016, OpenAI Gym, CoRR,.
11) R. Sawada, K. Sato, T. Majima, 2020, Automatic Ship Collision Avoidance using Deep Reinforcement Learning with LSTM in Continuous Action Spaces, Journal of Marine Science and Technology, https://doi.org/10.1007/s00773-020-00755-0.
12) R. Sawada, 2019, Automatic Collision Avoidance Using Deep Reinforcement Learning with Grid Sensor, Proceedings of the 23rd Asia Pacific Symposium on Intelligent and Evolutionary Systems (IES 2019), pp. 17-32.
13) 間島隆博，南真紀子，澤田涼平，福戸淳司：2019，自動避航操船の計算アルゴリズムの開発，第 19 回海技研研究発表会
14) 野本謙作・田口賢士：船の操縦性に就いて(2)，造船協会論文集，第 101 号, pp.57-66, 1956