2009 International Symposium on Nonlinear Theory and its Applications
NOLTA'09, Sapporo, Japan, October 18-21, 2009

# A Reinforcement Learning Approach to Course Decision of Ships under Navigation Rules

Takeshi Kamio[†], Shohei Sugeo[†], Kunihiko Mitsubori[††], Takahiro Tanaka[†††],
Chang-Jun Ahn[†], Hisato Fujisaka[†], and Kazuhisa Haeiwa[†]

† Hiroshima City University, 3-4-1, Ozuka-higashi, Asaminami-ku, Hiroshima-shi, Hiroshima, 731-3194, Japan
†† Takushoku University, 815-1, Tatemachi, Hachioji-shi, Tokyo, 193-0985, Japan
††† Japan Coast Guard Academy, 5-1, Wakaba-cho, Kure-shi, Hiroshima, 737-8512, Japan
Email: kamio@info.hiroshima-cu.ac.jp

**Abstract–** The transportation by ship is very important in countries with seas and wide rivers. In the case of Japan, it accounts for about 40% of the domestic physical distribution and for 90% and more of the international physical distribution. Therefore, the course decision of ships is an important problem in the field of the marine engineering. However, the optimality of the course of ships and the interaction between the maneuvering actions of navigators have not been sufficiently discussed yet. We regard the multi agent reinforcement learning (RL), which is an important learning algorithm in the field of the artificial intelligence and the machine learning, as a useful tool to brisk up these discussions. In this paper, we propose the RL framework to decide the course of ships under the navigation rules.

## 1. Introduction

The course decision of ships before the actual navigation is an important problem in the marine engineering. The importance of this problem deeply relates to the value of ships as the transportation and the difficulty in retrying the maneuvering motion. This difficulty is caused by the following four characteristics of ships.

1) The dynamics is nonlinear.
2) There is no way to brake and go backward effectively.
3) The attitude is unstable at a low speed.
4) The control tower does not exit.

In the field of the marine engineering, the course decision of ships has been treated in the maneuvering simulation and the automatic operation, where the course has been given as a guideline which the ship should trace and the procedures to avoid the collisions between ships have been discussed. But, the optimality of the course of ships and the interaction between the maneuvering actions of navigators have not been sufficiently discussed yet.

We regard the multi agent reinforcement learning (RL), which is an important learning algorithm in the field of the artificial intelligence and the machine learning, as a useful tool to brisk up the above discussions. Our opinion is based on the following two reasons. The first reason is that the course by RL can easily satisfy the actual navigators since each agent of RL optimizes a sequence of the action through the repeat of trial and error, which is executed according to the natural passage of time. The second reason is that navigators should be modeled as the competitive-cooperative multi agent system since the control tower does not exit.

Although there are a few works [1], [2] related the course decision of ships by RL, they are very simple methods and are not suitable for the actual navigation. Therefore, we propose the RL framework to decide the course of ships under the navigation rules in this paper.

## 2. Simple application of reinforcement learning (RL) to course decision of ships

### 2.1. Model of ship maneuvering motion

We use a simple response model (i.e., KT model [3]) as the model of the ship maneuvering motion. Fig.1 shows the model in a bird's eye view. $O_S$ is the center in turning the ship's head and represents the ship's location (i.e., $O_S = (x, y)$). $\phi$ is the heading angle. $L$ is the ship's length. $\mathbf{v}_0$ is the forward velocity vector and its size is $V_0$. The dynamics of the ship maneuvering motion is given by,

$$T\ddot{\phi} + \dot{\phi} = K\delta,\ \ \dot{x} = V_0 \sin\phi,\ \ \dot{y} = V_0 \cos\phi\ , \tag{1}$$

where $\delta$ is the rudder angle. $K$ and $T$ are the parameters to characterize the ship maneuvering performance in still water. They are given by $K=K_0/(L/V_0)$ and $T=T_0\times(L/V_0)$, where $K_0$ and $T_0$ are the dimensionless parameters. Each ship has individual values of $K_0$ and $T_0$. We fix the parameters $K_0$, $T_0$, and $L$ at the corresponding values of the patrol vessel KOJIMA in Japan Coast Guard (i.e., $K_0$= 1.310, $T_0$=1.085, $L$=107[m]). Since we consider the course decision of ships in the limited sea area, $V_0$ is also fixed at the standard value (i.e., $V_0$=6.17[m/s]).

### 2.2. Model of sea area

Fig.2 shows the model of the sea area which is used as the stage of RL in this paper. In this model, we refer to a

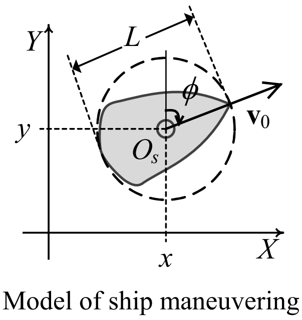


Fig.1 Model of ship maneuvering motion.

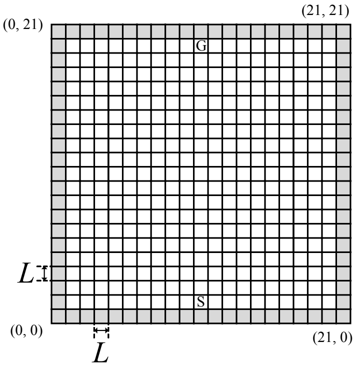


Fig.2 Model of sea area.

square as a grid and we fix its side length at $L$. Each grid is numbered for RL. There are four kinds of grids: normal one (white), no-entry one (gray), start one ("S"), and goal one ("G"). The no-entry grids represent the obstacles and the boundary where ships are permitted to move. Moreover, we assume that the tidal current does not affect the ship maneuvering motion.

### 2.3. Basic RL framework to decide course of ships

The Q-learning (QL) [4] is representative of RL. First, we explain the single agent QL. QL has the value function called Q-value, which is defined for each state-action pair. The aim of QL is getting Q-value to achieve a given task. QL is executed by iterating the following episodes. At the beginning of each episode, the environment is initialized (i.e., the agent is set to the starting point in each episode). After the agent senses the state $s_t \in S$ from the perceptual inputs $\mathbf{P}_t$ and selects the action $a_t \in A$ by the policy at the time $t$, the environment makes a transition to a new state $s_{t+1} \in S$ and gives the agent a reward $r_{t+1}$. In this case, Q-value $Q(s_t,a_t)$ is updated by

$$Q(s_t,a_t) \leftarrow (1-\alpha)Q(s_t,a_t) + \alpha\{r_{t+1} + \gamma \max_{a \in A} Q(s_{t+1},a)\} \qquad (2)$$

where $\alpha$ and $\gamma$ are the learning rate and the discount rate respectively. If $s_{t+1}$ corresponds to the terminal state (i.e., the agent arrives at the goal or failure), the present episode is finished and the next episode is started. These processes are iterated until the agent completes Q-value to maximize cumulative reward under given learning conditions. If the desired Q-value is obtained, the agent can achieve a given task. The policy used here is based on the ε-greedy policy. That is to say, while the agent basically selects the action with the largest Q-value in the current state, the action is randomly selected with a small probability $\varepsilon$. If $\varepsilon$=0, it is called the greedy policy.

Next, we consider a simple application of QL to the course decision of $n$-ships [2]. To achieve this, the single agent QL has to be expanded into the multi agent QL. Here, we show only the essentials. The ship $k$ (=1,···,$n$) is controlled by the agent $k$, which has its own Q-value. The aim of the agent $k$ is finding the best course for the ship $k$ to move from "S$^k$" to "G$^k$" without collisions. When the agent $k$ has obtained $Q^k(s^k_t,a^k_t)$ to maximize cumulative reward, the ship $k$ can move on the best course. The definitions of $s^k$, $a^k$, and $r^k$ are necessary for the agent $k$ to obtain such $Q^k(s^k_t,a^k_t)$. The agent $k$ senses the state $s^k_t$ from the perceptual inputs $\mathbf{P}^k_t$ (e.g., $\mathbf{P}^k_t$=($x^k_t$, $y^k_t$, $\phi^k_t$, $\dot{\phi}^k_t$)). Since QL cannot handle infinite states, each element of $\mathbf{P}^k$ has to be quantized. $x^k$ and $y^k$ are quantized by grids as shown in Fig.2. $\phi^k \in$[0°,360°] is divided into 12 equal parts and $\dot{\phi}^k$ is divided into 2 parts based on its sign. The total number of states equals 21×21×12×2=10584. Therefore, $s^k$ is given as an integer included in [0, 10583]. The action $a^k_t$ corresponds to one of 5 rudder angles {0°,10°,–10°,20°, –20°} and $a^k$ is given as an integer included in [0,4]. The reward $r^k$ depends on the grid in which the ship $k$ is. If the ship $k$ is in "G$^k$", no-entry grid, and the others, the agent $k$ receives $r^k$=1, $r^k$=–1, and $r^k$=0, respectively. However, if the ship $k$ and other ships are in a grid at the same time, the agent $k$ receives $r^k$=–1. The agent $k$ learns to avoid collisions by this interaction between ships. If all the ships arrive at their goal without collisions constantly, it means that the course decision of ships has been finished.

## 3. Reinforcement learning (RL) to decide course of ships under navigation rules

### 3.1. Collision situations and navigation rules

The navigation rules are the knowledge to avoid collisions between 2 ships, which actual navigators acquired through their experiences. They are provided by the international regulations [5]. In this paper, we introduce 3 navigation rules into QL for the course decision of ships. These rules correspond to 3 typical collision situations: Head-on-situation, Crossing Situation, and Overtaking.

Figs.3, 4, and 5 show the collision situations and the corresponding navigation rules. In the case of Head-on-situation shown in Fig.3(a), each ship must change the course to the right as shown in Fig.3(b). In the case of Crossing Situation shown in Fig.4(a), the ship which has the other ship on the right side must change the course to the right as shown in Fig.4(b). In the case of Overtaking shown in Fig.5(a), the overtaking ship must change the course to the right or the left as shown in Fig.5(b).

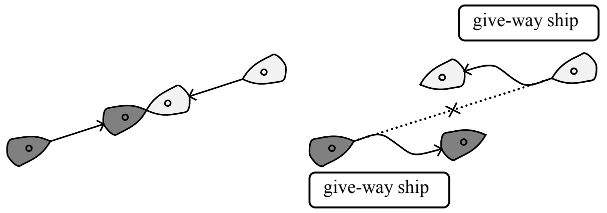
### 3.2. How to introduce navigation rules into RL for course decision of ships

To introduce the navigation rules into QL for the course decision of ships, the agent $k$ has to detect the ship $l$ ($\neq k$) in the view of the ship $k$, judge the collision situation of the ship $k$, and restrict the course of the ship $k$. We implement these requirements as follows.

The view of the ship $k$ is defined as the circle of radius $W_k$ centered at the ship $k$. If $W_k$ is larger than the distance ($d_{kl}$) between ships $k$ and $l$, the agent $k$ can detect the ship $l$, and vice versa, as shown in Fig.6.

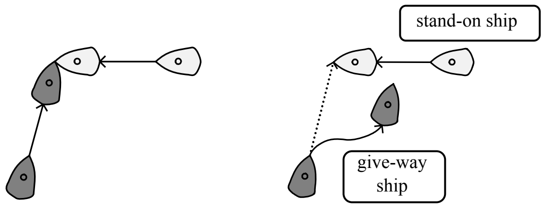
If the agent $k$ detects the ship $l$, it has to judge the collision situation from the position relation between ships $k$ and $l$. As shown in Fig.7, the agent $k$ can judge the collision situation by the angle $A^j_{kl}$, which is the direction of the ship $k$ from the head of the ship $l$. For example, if the ship $k$ is in $A^j_{kl} \in$[247.5°, 360°–$A^h_{kl}$], it is a give-way ship in Crossing Situation; similarly if the ship $k$ is in $A^j_{kl} \in$[0°, $A^h_{kl}$] or [360°–$A^h_{kl}$, 360°], it is a give-way ship in Head-on-situation. $A^h_{kl}$ is the angle to judge that ships $k$ and $l$ are in Head-on-situation, and it is calculated by

$$A^h_{kl} = | 2 \times \sin^{-1}\{H/(2d_{kl})\} | . \qquad (3)$$



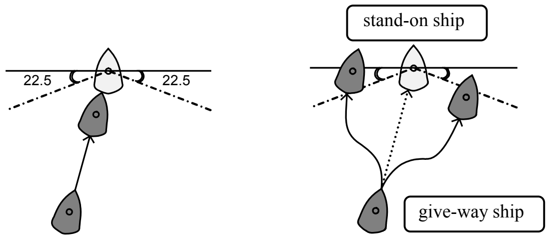
(a) collision (b) navigation rule

Fig.3 Head-on-situation.



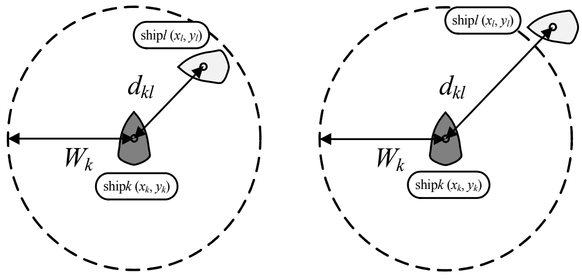
(a) collision (b) navigation rule

Fig.4 Crossing Situation.



(a) collision (b) navigation rule

Fig.5 Overtaking.



(a) $d_{kl}<W_k$ (b) $d_{kl}\geq W_k$

Fig.6 View of ship $k$.

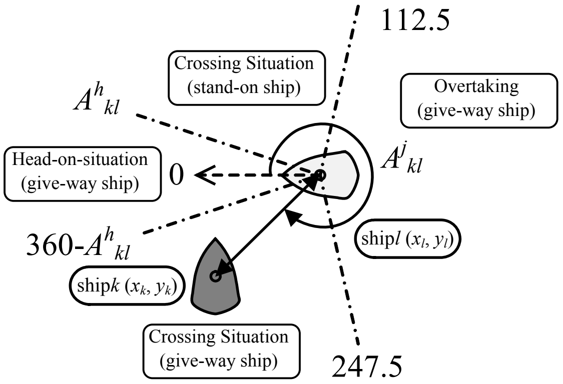

112.5
Crossing Situation (stand-on ship)
Overtaking (give-way ship)
$A^h_{kl}$
Head-on-situation (give-way ship)
0
$A^j_{kl}$
360-$A^h_{kl}$
ship$l$ ($x_l$, $y_l$)
ship$k$ ($x_k$, $y_k$)
Crossing Situation (give-way ship)
247.5

Fig.7 Judgment of collision situations.

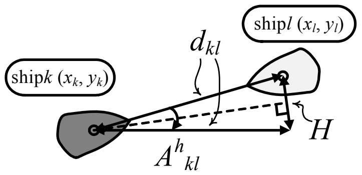
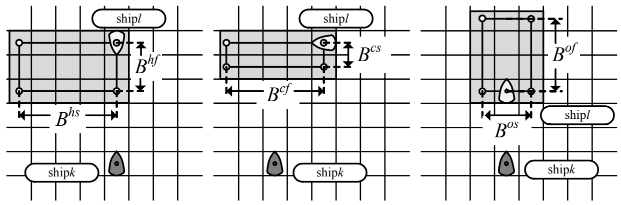


Fig.8 Angle for judgment of Head-on-situation.



(a) Head-on situation (b) Crossing Situation (c) Overtaking

Fig.9 Dynamic no-entry grids added around ship $l$.

As shown in Fig.8, the distance $H$ is needed when the ship $k$ meets the ship $l$.

If the agent $k$ judges that the ship $k$ should avoid the ship $l$ (i.e., the ship $k$ is a give-way ship), the dynamic no-entry grids are added around the ship $l$ as shown in Fig.9. These grids are based on the navigation rules. If the ship $k$ enters them, the agent $k$ receives the reward $r^k$=–1. As a result, the course of ship $k$ can be restricted appropriately.

### 3.3. Expanded RL framework to decide course of ships

Here, we propose a novel QL framework for the course decision of ships under the navigation rules. Our method consists of 2 stages.

The first stage is the process to decide the temporary course of each ship $k$ (=1,···,$n$). The temporary course is the course which each agent searches by the single agent QL, neglecting the other ships. When all the agents obtain their temporary courses, the first stage is completed. The aim of this stage is to decrease impractical collisions in the course decision of ships.

The second stage is the process to decide the final course of all the ships under the navigation rules. We apply the multi agent QL similar to Sect.2.3 to this stage. Before starting the second stage, the agent $k$ inherits $Q^k(s^k_t,a^k_t)$ from the first stage. At the beginning of each episode, each ship $k$ departs from "S$^k$" for "G$^k$" and the agent $k$ uses the greedy policy (i.e., $\varepsilon$=0). Each agent $k$ always observes whether other ships are in the view $W_k$ or not. If the agent $k$ detects the ship $l$ ($\neq k$) in the view (i.e., $d_{kl}<W_k$), the agent $k$ judges the collision situation by the criterion shown in Fig.7. If the ship $k$ is a give-way ship, the dynamic no-entry grids are added around the ship $l$ as shown in Fig.9. These added grids affect only the ship $k$. After detecting the ship $l$, the agent $k$ uses the ε-greedy policy until the present episode is finished. If the ship $k$ enters the grids added around the ship $l$, the agent $k$ receives the reward $r^k$=–1 and the ship $k$ is removed from the sea area. On the other hand, the ship $l$ receives no penalty. Also, if any ship enters the no-entry grids fixed in the sea area, the agent receives the reward $r$=–1 and the ship is removed from the sea area. If the ship $k$ arrives at the goal, the agent $k$ receives the reward $r^k$=1 and the ship

*k* is removed from the sea area. When all the ships are removed from the sea area, the present episode is finished. If all the ships arrive at their goal without collisions constantly, it means that our method completes the course decision of ships under the navigation rules.

## 4. Simulation Results

Simulations have been carried out to confirm that our proposed method can decide the course of ships under the navigation rules appropriately. Each ship has the common parameters except the size of the velocity. The common parameters are as follows: $\alpha$=0.2, $\gamma$=0.9, $\varepsilon$=1.0×10$^{-7}$, $W$= 10$L$, $H$=$L$, $B^{hf}$=2$L$, $B^{hs}$=9$L$, $B^{cf}$=10$L$, $B^{cs}$=$L$, $B^{of}$=5$L$, $B^{os}$=2$L$. The overtaking ship has 3$V_0$ and the others have $V_0$.

Figs. 10, 11, and 12 show the courses corresponding to 3 typical collision situations between 2 ships: Head-on-situation, Crossing Situation, and Overtaking. Fig.13 show the course corresponding to a complex collision situation between 4 ships. Each course is emphasized every 250 time-steps ($\Delta t$=0.2[s]) of numerical integration by the corresponding mark. From these simulation results, we have confirmed that each temporary course has the predicted collision and each final course has no collision and obeys the navigation rules appropriately.

Finally, we discuss the convergence of our learning method. Generally the convergence of QL is guaranteed by the assumption that the environment does not change. Although our learning method changes the environment by the dynamic no-entry grids in the second stage, we have not observed the convergence problem in these simulations. From these facts, we think that the temporary courses decided in the first stage could stabilize both the position and existence duration of the dynamic no-entry grids to a great extent; as a result, the presence of the dynamic no-entry grids did not cause serious damage to the convergence of our learning method.

## 5. Conclusions

We have proposed the RL framework to decide the course of ships under the navigation rules corresponding to 3 typical collision situations (Head-on-situation, Crossing Situation, and Overtaking). Simulation results have shown that our method can obtain an appropriate course which obeys the navigation rules. In the future, we will estimate our method as an assessment tool for the sea traffic.

**Acknowledgments**

This research was partially supported by the Ministry of Education, Science, Sports and Culture, Grant-in-Aid for Young Scientists (B), 21700172, 2009-2010.

## References

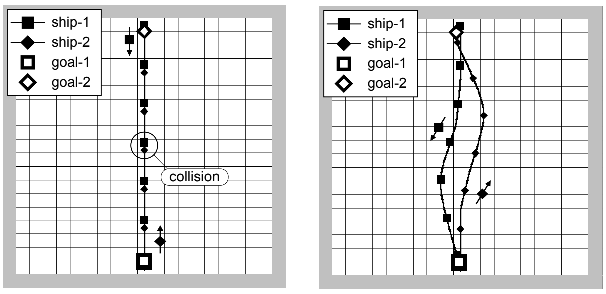
[1] T. Horiuchi, A. Fujino, O. Katai, and T. Sawaragi, "Q-PSP learning: an exploitation-oriented Q-learning algorithm and its application," Trans. of Society of Instrument and Control Engineers, vol.35, no.5, pp.645-653, 1999 (in Japanese).

[2] K. Mitsubori, T. Kamio, and T. Tanaka, "Finding the shortest course of a ship based on reinforcement learning algorithm," Journal of Japan Institute of Navigation, 110, pp.9-18, 2004.

[3] T. I. Fossen, "Guidance and Control of Ocean Vehicle," John Wiley & Sons Ltd., pp.172-174, 1994.

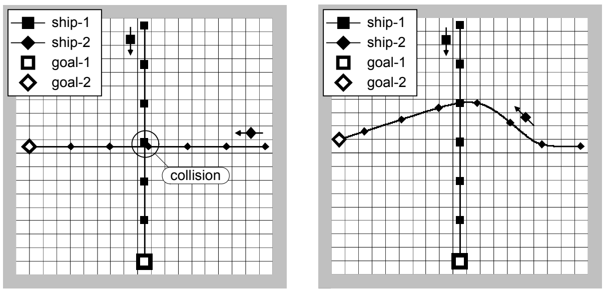
[4] C. J. C. H. Watkins and P. Dayan, "Q-learning", Machine Learning, 8, pp.279-292, 1992.

[5] International Maritime Organization, "International Regulations for Preventing Collisions at Sea," 1972.



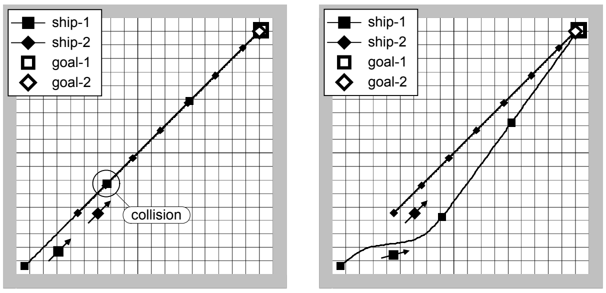
(a) temporary course (b) final course

Fig.10 Head-on-situation.



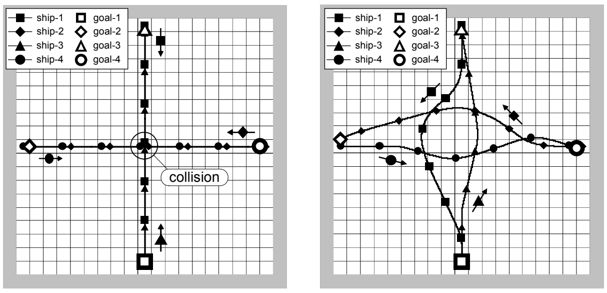
(a) temporary course (b) final course

Fig.11 Crossing Situation.



(a) temporary course (b) final course

Fig.12 Overtaking.



(a) temporary course (b) final course

Fig.13 Collision between 4 ships.