International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

Multi-Ship Collision Avoidance Method based on Markov Decision Process

Yousra Melhaoui*[1], Abdelali Kamil[2], Khalifa Mansouri[3], Mostafa Rachik[4]

Submitted: 18/11/2022 **Revised**: 15/01/2023 **Accepted**: 14/02/2023

Abstract: The maritime industry has been incorporating advanced technology to enhance mission planning and ensure safe navigation, including autonomous collision avoidance systems that follow International Regulations for Preventing Collisions at Sea (COLREG). Ongoing research in this field encompasses a wide range of approaches, from optimal control analysis to heuristic and metaheuristic methods, and solutions based on artificial intelligence. In this study, we propose an autonomous collision avoidance algorithm for ships based on Markov decision process. This work focuses on the development of a COLREG-compliant autonomous collision avoidance algorithm for ships using a Markov decision process. The algorithm considers the subject ship's position and aims to resolve potential collision conflicts with target ships while keeping the vessel on its initial trajectory, in compliance with regulations. The system is modeled as a Markov decision process using the ship's three coordinates position as states, actions generated from degrees-of-freedom, and constraints such as safe path, trip cost, and respect for rules to design the reward. The proposed policy search algorithm is implemented using python and its convergence and efficiency are tested through multiple scenarios.

Keywords: *autonomous navigation, collision avoidance, colreg rules, decision-process, grid-world, markov property, optimization, policy search, safe trajectory, ship motion, value iteration.*

1. Introduction

Maritime navigation integrates sophisticated technology, including mission planning systems, guidance, navigation, and control systems, such as path and trajectory tracking, dynamic weather routing, and dynamic positioning. These technologies form a major foundation for realizing future expectations for the deployment of commercial autonomous ships. To ensure safe maneuvering, keep the planned trajectories, and comply with the core of International Regulations for Preventing Collisions at Sea (COLREG), both academia and industry are investing in research and development of autonomous collision avoidance systems. Different approaches have been proposed to improve maritime safety, including the analysis of the problem as an predictive control problem [1], [2] and, [3], the use of stochastic techniques or heuristic and metaheuristic algorithms [4], [5] and, [6], and the integration of artificial intelligence [7].

The use of Markov Decision Process (MDP) has been established as the preferred framework for addressing the optimal control of autonomous collision avoidance systems that aims to simultaneously maximize safety and minimize trajectory cost. The efficacy of MDP in solving sequential decision-making problems in unmanned aircraft has been demonstrated in previous studies. The collision avoidance problem is formulated as an MDP problem to balance both the risk of collision and the cost of deviation in unmanned aircraft [8]. The proposed approach in [9] adds a decomposition and coordination mechanism to MDP, utilizing closest threat heuristics and an uncoordinated algorithm to resolve multi-aircraft conflicts. The research in [10] addresses the computational complexity of higher-dimensional and uncontrollable aspects in the conflict environment of unmanned airborne collision avoidance, which is formulated as a Markov Decision Problem . A decision-based strategy is proposed to counteract delayed remote human pilot commands and control collision avoidance in unmanned aircraft [11]. The method optimizes the decision between generating a safe path and taking evasive action, or waiting for the pilot command, and provides an effective solution to the computational challenges of the collision avoidance problem. A study employs a gridding system to construct an algorithm that generates multiple threat resolutions for autonomous collision avoidance in unmanned aerial vehicles [12]. Previous studies have demonstrated the advantages of using MDP-based problem formulation and Q-learning to address COLREG rules and multiple actors in ship collision avoidance problems [13]. This study proposes and develops a collision avoidance framework that adheres to COLREG rules and employs a Q-learning algorithm to ensure safe navigation of unmanned ships. In [7], the author proposes a method that combines the asynchronous advantage actor-critic (A3C) algorithm, a long short-term memory neural network (LSTM), and Q-learning to overcome the low performance issue of model-free reinforcement learning in multi-ship collision avoidance under unknown

[1] *Laboratory LAMS, Faculty of Sciences Ben M'sik, Hassan II University, Casablanca, Morocco*
ORCID ID: 0000-0002-7573-3458
[2] *Laboratory LAMS, Faculty of Sciences Ben M'sik, Hassan II University, Casablanca, Morocco*
ORCID ID: 0000-0002-6126-7345
[3] *Laboratory SSDIA, ENSET Mohammedia, Hassan II University, Casablanca, Morocco*
ORCID ID: 0000-0003-2242-0973
[4] *Laboratory LAMS, Faculty of Sciences Ben M'sik, Hassan II University, Casablanca, Morocco*
ORCID ID: 0000-0002-5118-2786
** Corresponding Author Email: yousra.melhaoui.1@gmail.com*

environments. Other studies have demonstrated the effectiveness of MDP in solving a variety of problems. For example, [14] introduces linear temporal logic with MDP to develop a dynamic programming algorithm that generates an optimal policy for robotic applications. The authors formulate the state and cost function using MDP and use linear temporal logic to formulate constraints, aiming to solve dynamic programming problems by providing an optimal policy that meets a set of specifications. Another study applies the Markov Decision Process to control and optimize the management of human resources in a multi-level hierarchical system, taking into account different types of state transitions [15].

The aim of this paper is to advance the development of an autonomous collision avoidance system. The proposed algorithm solves the collision conflict between the subject ship and a target ship while adhering to COLREG rules and maintaining the initial trajectory. The system is formulated as a Markov Decision Process, using the three coordinates of ship position in a limited encounter as states, degrees-of-freedom as actions, and constraints such as safe path, trip cost, and COLREG compliance to design rewards and policies driven by the rudder and thrusters controls. Previous studies have used MDP to solve ship collision avoidance, but only considered the subject ship as the agent. This paper extends the MDP formulation to a dynamic collision avoidance system, modifying the reward design to account for the actions of target ships and manage multi-ship collision risk in real-time. A value iteration-based policy search algorithm is presented and implemented using Python. The algorithm's convergence is verified, and its efficiency is evaluated through multiple scenario tests.

The structure of this paper is outlined as follows. Section I presents the background and motivation of the study. In Section II, the mathematical model for ship motion and the safe trajectory in a multi-ship environment, as well as the formulation of COLREG collision avoidance rules, are discussed. The formulation of the MDP problem and the policy search using value iteration are presented in Section III. Section IV provides an explanation of the proposed collision avoidance algorithm and demonstrates its effectiveness through a multi-ship collision avoidance use case and numerical simulation results. Finally, the paper concludes by summarizing the main results of the research.

2. System Description

2.1 Ship Motion

The ship exhibits horizontal plane motion, typically represented in maneuvering and control models by three degrees of freedom (DOF) equations describing its three motions in a rigid body frame $\boldsymbol{O_B(x_B, y_B, z_B)}$ as shown in **Fig. 1**.

- Surge, translation following $\boldsymbol{x}$ axis.
- Sway, translation following $\boldsymbol{y}$ axis.
- Yaw, rotation about $\boldsymbol{z}$ axis.

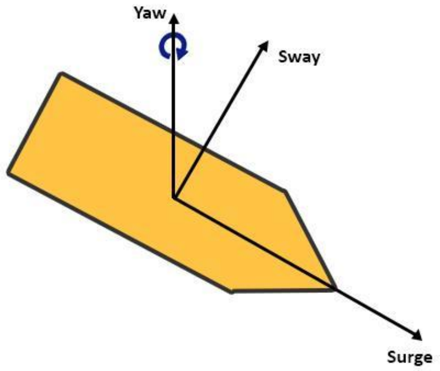


Fig. 1. Three degrees of freedom of a ship.

The motion can be expressed as a vector in the earth-fixed frame $\boldsymbol{O(x, y, z)}$ [16], [17].

$$\begin{cases} \dot{\boldsymbol{\eta}} = \boldsymbol{R(\psi)\nu} \\ \boldsymbol{M\dot{\nu} + C(\nu)\nu + D(\nu)\nu = \tau} \end{cases}$$

The state of the ship in the earth-fixed frame can be represented by $\boldsymbol{\eta} = [\boldsymbol{x\ \ y\ \ \psi}]^{\boldsymbol{T}}$. The velocity vector is denoted as $\boldsymbol{\nu} = [\boldsymbol{u\ \ v\ \ r}]^{\boldsymbol{T}}$, while the mass matrix, Coriolis matrix, and damp matrix are represented by $\boldsymbol{M}$, $\boldsymbol{C}$, and $\boldsymbol{D}$ respectively. The actuators forces and moments are represented by $\boldsymbol{\tau}$. $\boldsymbol{I_Z}$ represents the inertial moment about the first component of the center gravity $(\boldsymbol{x_g, y_g, z_g})$,

$$\boldsymbol{R(\psi)} = \begin{pmatrix} \cos(\psi) & -\sin(\psi) & 0 \\ \sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$\boldsymbol{M} = \begin{pmatrix} m - X_{\dot{u}} & 0 & 0 \\ 0 & m - Y_{\dot{v}} & mx_g - Y_{\dot{r}} \\ 0 & mx_g - N_{\dot{v}} & I_z - N_{\dot{r}} \end{pmatrix}$$

$$\boldsymbol{C} = \begin{pmatrix} 0 & 0 & -m(x_g r + v) + Y_{\dot{v}} v + Y_{\dot{r}} r \\ 0 & 0 & mu_0 - X_{\dot{u}}\, u_0 \\ -Y_{\dot{v}} v N_{\dot{v}} r & -mu_0 + X_u u_0 & 0 \end{pmatrix}$$

$$\boldsymbol{D} = \begin{pmatrix} X_u - X_{\{u|u\}}|u| & 0 & 0 \\ 0 & -Y_v - Y_{v|v}|v| - Y_{r|v}|r| & -Y - Y_{r|r}|r| - Y_{v|r}|v| \\ 0 & -N_v - N_{v|r}|v| - N_{r|v}|r| & -N_r - N_{r|r}|r| - N_{v|r}|v| \end{pmatrix}$$

$\boldsymbol{\tau}$ denotes the actuator forces and moments [18].

2.2 Safe Ship Trajectories

The simplest form of the ship's trajectory is represented as a time-varying state $\boldsymbol{\eta(t)} = [\boldsymbol{x(t)\ y(t)\ \ \psi(t)}\ ]^{\boldsymbol{T}}$. To ensure the ship's safety and avoid any collision risk with other vessels, its trajectory should belong to a set of safe ship trajectories $\boldsymbol{C_0(t)}$ [19], which is defined as:

$$\boldsymbol{C_{\{0\}}(t)} = \{\, \boldsymbol{\eta_{\{0\}}}\ \boldsymbol{(t)}\ \ /\ \ \boldsymbol{\eta_{\{0\}}(t)} \cap\ \boldsymbol{\eta_{\{i\}}(t)} = \emptyset\ \ ;\ \forall\ \boldsymbol{i} \in \boldsymbol{N}\, \}$$

Where $\boldsymbol{N}$ is the number of target ships detected by Automatic Identification System (AIS).

The maritime safety protocol requires ships to maintain a safe distance, $\boldsymbol{D^i_{safe}}$, to prevent collisions [20]

$$\left(\boldsymbol{x_0(t) - x_0(t)}\right)^2 + \left(\boldsymbol{y_0(t) - y_i(t)}\right)^2 \geq\ \boldsymbol{D^2_{safe}}$$

$$\boldsymbol{D^i_{safe}} = \boldsymbol{R^i} + \boldsymbol{D^i_0} + \frac{\boldsymbol{L}}{\boldsymbol{2}}$$

- $\boldsymbol{R^i}$: The domain radius of the $\boldsymbol{i^{th}}$ target ship.
- $\boldsymbol{L}$: The subject ship length.
- $\boldsymbol{D^i_0}$: The safety distance between the target ship and the subject ship described in COLREG rules.

Subsequently, consider.

$$\boldsymbol{d(\eta_0(t), \eta_i(t))} = \left(\boldsymbol{x_0(t) - x_i(t)}\right)^2 + \left(\boldsymbol{y_0(t) - y_i(t)}\right)^2$$

The safe ship trajectories can be formulated by respecting the safety distance D^{i}_{safe} to minimize collision risk between the subject ship and target ships [19].

$$C_{\{0\}}(t) = \{\eta_{\{o\}}(t) \;/\; d\left(\eta_{\{o\}}(t), \eta_{\{i\}}(t)\right) \geq D^{\{i\}}_{\{safe\}};\; \forall\; i \in\; N\}$$

2.3 COLREG Rules

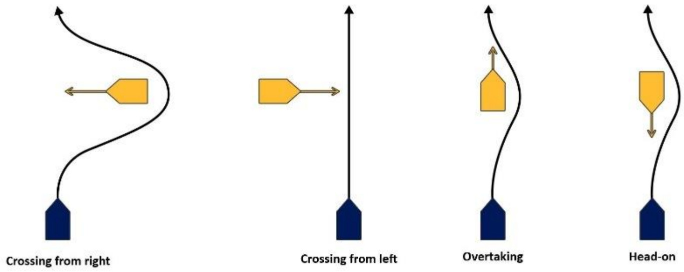


Fig. 2. Maneuvers required for various COLREG situations.

The COLREG rules address collision avoidance in three scenarios, as depicted in **Fig. 2.** The heading angles of the subject ship and the target ship are represented by ψ_0 and ψ_i respectively.

- In the HEAD-ON situation as per COLREG rules, each ship must alter its course to starboard (right). This situation is determined when the angle between the target ship and the subject ship satisfies the conditions illustrated in **Fig. 3**.

$$\frac{6\,\pi}{8} \leq |\,\psi_0 - \psi_i| \leq \frac{10\,\pi}{8}.$$

- In the CROSSING situation, the ship on the port side should alter its course to starboard and the other ship should maintain its course. This situation is determined by the angle between the target ship and the subject ship, as shown in **Fig. 3**.

$$\frac{2\,\pi}{8} \leq |\,\psi_0 - \psi_i| \leq \frac{6\,\pi}{8} \quad ou \quad \frac{10\,\pi}{8} \leq |\,\psi_0 - \psi_i| \leq \frac{14\,\pi}{8}$$

- In the OVERTAKING situation, the ship that is being overtaken should alter to starboard and the overtaking ship should maintain its course. This scenario is determined by the inequality shown in **Fig. 3**.

$$|\,\psi_0 - \psi_i| \leq \frac{2\,\pi}{8} \quad ou \quad |\,\psi_0 - \psi_i| \geq \frac{14\,\pi}{8}$$

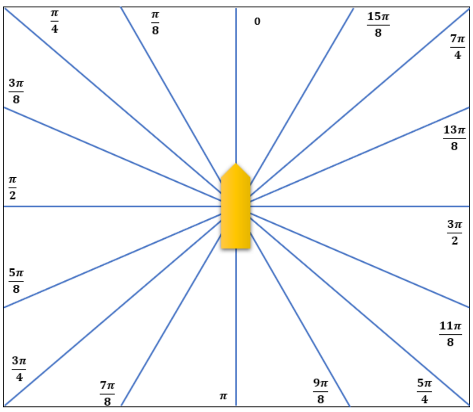


Fig. 3. Cap angles design.

3. MDP Implementation

3.1 Problem Formulation

The collision avoidance problem with multiple ship is a sequential decision process which is defined as a dynamic system model controlled by a decision maker [21]. The decision process depends on:

- The time epochs of decision making,
- The length of the decision-making horizon,
- The mathematical properties of the state and action spaces,
- The optimality criteria

Furthermore, in each time step, the decision maker must assess the current states of the subject ship and target ships to choose the optimal action based on their expertise and the COLREG rules. The current selected action is not dependent on the history of previous actions. The system's evolution is stochastic and history-independent, thus the collision avoidance process can be modeled as a Markov Decision Process [22].

The proposed MDP-based multi-ship collision avoidance system is designed to minimize the risk of collision between ships, which are treated as agents. The system is shown in **Fig. 4** and can be represented as a tuple, denoted as $\langle S,\ A,\ P,\ R,\ \gamma \rangle$:

- S : The set of states is defined by the distinction between safe actor ship trajectories $\eta_{safe}(t) \in C_0(t)$ and unsafe actor ship trajectories $\eta_{unsafe}(t) \notin C_0(t)$:

$$S = C_0(t) \cup \overline{C_0(t)}$$

- A: The set of actions available in the system is determined by the longitudinal speed and the change in cap angle, represented as $\{\mu = (n, \delta)\}$. These actions include Up, Up Right, Right, Down Right, Down, Down Left, Left, and Up Left, as well as a No Action option to comply with COLREG rules, that requires the ship to maintain its course and speed in some cases. The finite set of actions is expressed as,

$$A = \{U, UR, R, DR, D, DL, L, UL, NA\}$$

- $P: S \times A \times S \rightarrow [0\ 1]$: The transition from the current state s to the next state s' is characterized by a probability density function $P(s'|s, a)$ that represents the conditional distribution of the next state, given the current state and the selected action a. This dependence is in accordance with the Markov property, meaning that it only depends on the current state and action and not on any previous states or actions.

- R : The reward function evaluates the action taken by an actor ship, with the aim of optimizing its strategy to maximize the return. The next action selection is dependent on the outcome of the reward function.

- γ : The discount factor is a scalar in the interval $[0, 1]$ that represents the relative importance of future rewards in determining the present-time action selection strategy. The use of a discount factor ensures the convergence of the algorithm by factoring in the effect of future rewards on the present decision-making process.

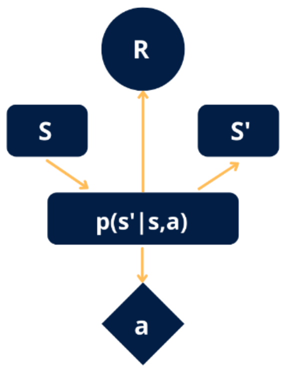


Fig. 4. Markov decision process schema.

3.2 Reward Design

The Markov decision process utilizes rewards to evaluate the effectiveness of actions taken by agents in resolving problems. In the case of collision avoidance, non-negative rewards are constructed based on states and actions by incorporating three desired behaviors: avoiding collisions, compliance with rules, and cost minimization.

First, we define the reward for collision avoidance $\boldsymbol{\Delta_{col}}$ as:

$$\begin{cases} \Delta_{col} = +0.25 & if\ \eta_0 \in C_0(t) \\ \Delta_{col} = -0.25 & else \end{cases}$$

There exist two cases to define the reward for respecting rules, depending on colreg rules:

Case 1

The subject ship is on the port, which means it should take actions to the starboard:

$$0 \le |\psi_{\{0\}} - \psi_i| \le \frac{10\,\pi}{8} \quad or\ |\psi_0 - \psi_{\{i\}}| \ge \frac{14\,\pi}{8}$$

Then:

$$\begin{cases} \Delta_{rules} = +0.25 & a(s(t)) \in \{UR, DR, R\} \\ \Delta_{rules} = 0 & else \end{cases}$$

Case 2

The subject ship is on the starboard side, which means it should not act:

$$\frac{10\,\pi}{8} \le |\psi_{\{0\}} - \psi_i| \le \frac{14\,\pi}{8}$$

Then:

$$\begin{cases} \Delta_{rules} = +0.25 & a(s(t)) \in \{U, D, NA\} \\ \Delta_{rules} = 0 & else \end{cases}$$

To guarantee the selection of the short safe path, we define the reward for minimizing cost that can be expressed as:

$$\begin{cases} \Delta_{cost} = -0.25 & \|\eta_0(t_{final}) - \eta_0(t+1)\| \ge \|\eta_0(t_{final}) - \eta_0(t)\| \\ \Delta_{cost} = +0.25 & else \end{cases}$$

The proposed total reward function is defined as the sum of three rewards previously defined: collision avoidance, respecting rules, and minimizing cost.

$$\Delta = \Delta_{col} + \Delta_{rules} + \Delta_{cost}$$

The cumulative reward function from a state $\boldsymbol{s}$ and action $\boldsymbol{a}$ over a finite horizon $\boldsymbol{T}$ is defined as the discounted sum of future rewards, as follows:

$$R(s,a) = \sum_{k=0}^{T-1} \gamma^k \Delta_k$$

3.3 Optimal Policy Search Method

The decision maker selects an action from the available actions set $\boldsymbol{A}$ based on the system state observation at each time step when a decision is made. The policy function maps actions to states sets and is denoted by $\boldsymbol{\Pi}$. The set of non-stationary Markovian policies, is defined as $\boldsymbol{\pi} = \{\boldsymbol{\pi_t}, \boldsymbol{t} = 0, 1, \ldots\}$, where $\boldsymbol{\pi_t}: \boldsymbol{S} \to \boldsymbol{A}$ \$ is a function that maps states to actions such that $\boldsymbol{\pi_t}(\boldsymbol{s}) \in \boldsymbol{A}(\boldsymbol{s})$ for each $\boldsymbol{s} \in \boldsymbol{S}$ [21]. The objective of the MDP is to find the optimal policy π^* that maximizes the expected discounted cumulative reward. The proposed algorithm trains the ship actor to learn a policy as close as possible to the optimal policy. To evaluate the next state, a value function is determined, which associates a real number with each state. The value function, expressed as the expected discounted cumulative reward for a ship actor starting from a state $\boldsymbol{s}$ [23], is defined as:

$$V^{\pi}(s) = \mathrm{E}\left[\sum_{k=0}^{T-1} \gamma^k \Delta_k\right]$$

A state where the ship is safe, follows COLREG rules, and remains on its original trajectory is rated highly. The optimal value function $\boldsymbol{V}^*(\boldsymbol{s})$ maximizes the value function and results in a higher score. The optimal policy function is then defined as $\boldsymbol{\pi}(\boldsymbol{s}) = \boldsymbol{argmax}\, \boldsymbol{V}^{\boldsymbol{\pi}}(\boldsymbol{s})$. The Bellman Equation helps determine the value function and reach the goal [23]. The Q-value function, defined as the expected total discounted reward starting from state $\boldsymbol{s}$ and selecting action $\boldsymbol{a}$, is a useful tool in this regard and is expressed as $\boldsymbol{Q}(\boldsymbol{s}, \boldsymbol{a}) = \boldsymbol{max}\, \boldsymbol{V}(\boldsymbol{s})$. The optimal Q-function $\boldsymbol{Q}^*(\boldsymbol{s}, \boldsymbol{a})$ is given explicitly by:

$$Q^*(s,a) = \Delta(s,a) + \gamma \sum_{s'} P(s'|s,a) \times V^*(s')\,.$$

Then, the Bellman equation is written as following:

$$V^*(s) = \max_a [\Delta(s,a) + \gamma \textstyle\sum_{s'} P(s'|s,a) \times V^*(s')]\ .$$

Using the value iteration method to solve the problem, this method calculates the optimal value function $\boldsymbol{V}^*(\boldsymbol{s})$ by iteratively improving the estimating $\boldsymbol{V}^{\boldsymbol{\pi}}(\boldsymbol{s})$.

4. Simulation and Numerical Results

4.1 MDPCA Algorithm

The initial states and actions of the markov decision process collision avoidance algorithm (MDPCA) are defined based on pre-established transition mappings. The number $\boldsymbol{N}$ of target ships in the encounter is determined through the use of Automatic Identification System (AIS) technology [24]. The speed and safe distances $\boldsymbol{D^i_{safe}}$ are calculated to determine the final time $\boldsymbol{t_{final}}$, which is the moment when the subject ship is no longer at risk of collision and has returned to its initial trajectory.

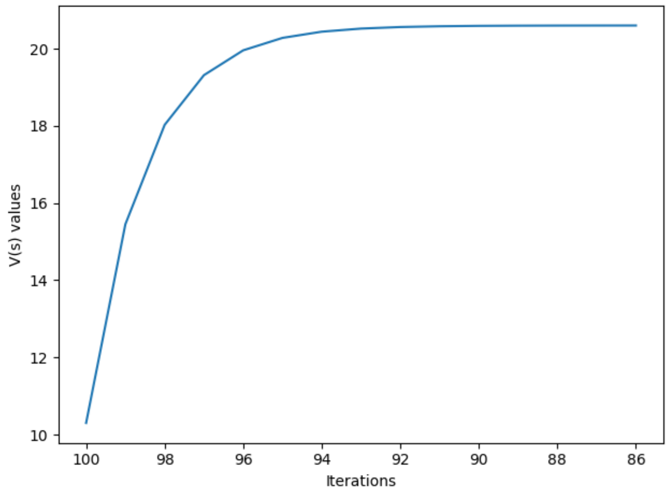


Fig. 5. Convergence of the algorithm in a finite number of steps.

Considering the dynamic nature of target ships, their positions, and distances from the subject ship change over time. Hence, the optimal policy must be recalculated for each state at every time step, considering the updated knowledge of positions and distances. The time interval is discretized into multiple constant steps. At each step, the subject vessel calculates the positions and distances from target ships. The algorithm updates the **Q(s, a)** and **V(s)** values until the convergence criteria are met as observed in **Fig. 5**. The resulting optimal policy is appended to the sequence of optimal policies.

Algorithm: Markov decision process collision avoidance.

initialize:

states ← from the transitions function.

actions ← from the transitions function.

subject_vessel_initial_position ← real position of the subject ship.

N **← number of encounter target ships.**

for $t \in [t_0\, t_{final}]$**:**

initialize:

target_positions ← dictionary of target ship: position.

do

for all s states:

for all a actions:

$$Q(s,a) \leftarrow \Delta(s,a) + \gamma \sum_{s'} P(s'|s,a) \times \; V^*(s').$$

$$V(s) \leftarrow max\, Q(s,a)$$

until V(s) converges

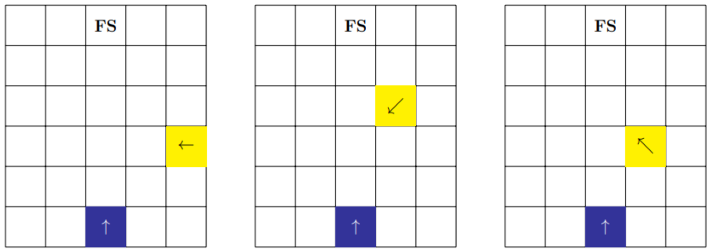
optimal_policy ← $argmax\ Q(s,a)$

optimal_policies_sequence ← add optimal_policy to the sequence.

The objective of this section is to represent a simulation of the algorithm to adequately demonstrate the collision avoidance ability and effectiveness of the proposed method. The development of the algorithm is carried out using Python programming language via the PyCharm platform.

Based on the study in [25], a collision avoidance algorithm can be demonstrated in a static grid environment. The environment is illustrated by a 6x5 cell rectangular grid, as shown in **Fig. 6**. and **Fig. 7**. The movement of the ship is defined as XY plane. The ship can move to an adjacent grid at regular time intervals. The cells of the grid represent the ships states with three coordinates following x axis, y axis and the head angle ψ.

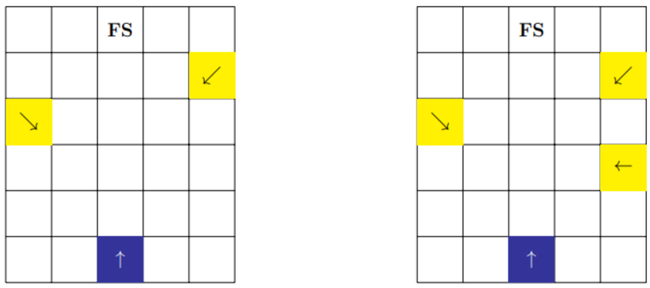
For each ship state, all actions in set A are possible, causing the agent to move one cell in the corresponding direction on the grid, as determined by the transition function. Given an action and an initial state, the function maps the next state to a probability value within the range of $[\mathbf{0}, \mathbf{1}]$, with the sum of values for a specific action and initial state equaling $\mathbf{1}$. The transition of the ship from one state to other results in a reward based on its position relative to a fixed target ship, while adhering to the colreg rules and minimizing cost, as outlined in the section on reward design. Target ships follow their own trajectories, represented as a sequence of states per time step. In each iteration, the reward for collision avoidance is calculated based on the detected state of target ships.



(a) One target in crossing case. (b) One target in head-on case. (c) One target in overtaken case.

Fig. 6. One-ship encounter COLREG cases.

The position of the subject vessel is evaluated in relation to an opposing target vessels in various configurations. Specifically, the position of the subject vessel is established at coordinates $(\mathbf{1}, \mathbf{3}, \mathbf{0})$ with a predetermined final destination of $(\mathbf{6}, \mathbf{3}, \mathbf{0})$ as can be observed **Fig. 6** and **Fig. 7**. The algorithm was tested using various encounter situations. Including easy one like one target per colreg collision case to difficult one like encounter of three targets with different colreg cases. The blue cell corresponds to the subject ship and the yellow cells to target ships.



(a) Two targets in crossing cases. (b) Three argets in rossing ases.

Fig. 7. Multi-ship encounter COLREG cases.

Case1

The three scenarios considered in the first case are the crossing, head-on, and overtaking situations, shown in **Fig. 6**. The algorithm allows the subject vessel to avoid collision with the target vessel in the fewest steps, as shown in **Fig. 8**, **Fig. 9**, and **Fig. 10**. The subject ship's proposed actions are shown with blue arrows and the target ships with yellow arrows. The algorithm ensures COLREG compliance with a starboard maneuver.

Fig. 8. One target in crossing case.

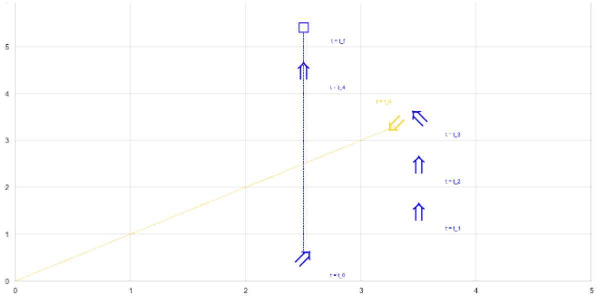


Fig. 9. One target in head-on case.

Fig. 10. One target in overtaken case.

Case 2

In **Fig. 7**, the subject vessel is shown to be in encounters with multiple ships. The figure illustrates two scenarios, with two and three target ships positioned differently. The algorithm demonstrates its effectiveness in avoiding collision with the target ships, as depicted in **Fig. 11**, **Fig. 12**, and **Fig. 13**. In case of cooperative target ship that follows the COLREG rules to avoid collision, the proposed maneuvers are still effective, as shown in **Fig. 12**. Furthermore, the algorithm abides by the COLREG rules by implementing a turn to the starboard side.

Fig. 11. Two targets in crossing case.

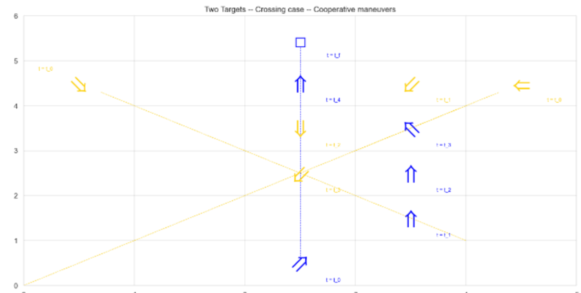


Fig.12. Two targets in cooperative crossing case.

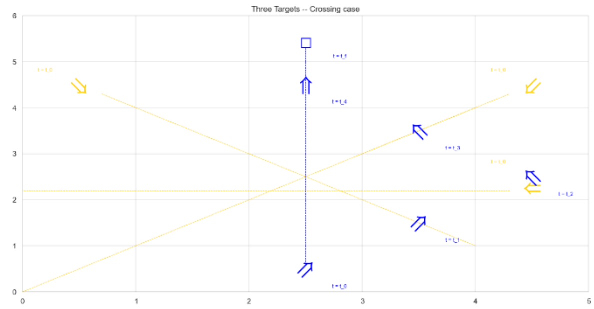


Fig. 13. Three targets in crossing case.

Conclusion

In conclusion, this paper presents a significant contribution to the development of an autonomous collision avoidance system. We proposed an algorithm that considers both the subject ship and target ships, and we designed a system that respect COLREG rules while keeping the ship on its initial trajectory, we advanced the field of collision avoidance for autonomous ships. We formulated the system as a Markov Decision Process, using three coordinates of ship position as the set of states, degrees-of-freedom as the set of actions, and constraints such as safe path, trip cost, and respect for COLREG rules to design rewards. Additionally, we implemented a policy search algorithm using Python, and verified its convergence and efficiency through multiple scenarios.

Author contributions

Yousra Melhaoui: Conception, Methodology, data analysis and interpretation, Writing-Original draft preparation, software, Field study. **Abdelali Kamil:** contributed data analysis tools, software, Field study. **Khalifa Mansouri:** Investigation, Writing-Reviewing and Editing, Validation. **Mostafa Rachik:** Investigation, Writing-Reviewing and Editing, Validation.

Conflicts of interest

The authors declare no conflicts of interest.

References

[1] A. Miller, “Model Predictive Control of the ship’s motion in presence of wind disturbances,” *Zesz. Nauk. Morska W Szczecinie*, no. 39 (111), pp. 107–115, 2014.

[2] M. Breivik, “MPC-based mid-level collision avoidance for ASVs using nonlinear programming,” presented at the 2017 IEEE Conference on Control Technology and Applications (CCTA), 2017, pp. 766–772.

[3] I. B. Hagen, D. K. M. Kufoalor, E. F. Brekke, and T. A. Johansen, “MPC-based collision avoidance strategy for existing marine vessel guidance systems,” presented at the 2018 IEEE International Conference on Robotics and Automation (ICRA), 2018, pp. 7618–7623.

[4] M.-C. Tsou and C.-K. Hsueh, “The study of ship collision avoidance route planning by ant colony algorithm,” *J. Mar. Sci. Technol.*, vol. 18, no. 5, p. 16, 2010.

[5] A. Lazarowska, “Swarm intelligence approach to safe ship control,” *Pol. Marit. Res.*, vol. 22, no. 4, pp. 34–40, 2015.

[6] Y. Zhao, W. Li, S. Feng, W. Y. Ochieng, and W. Schuster, “An improved differential evolution algorithm for maritime collision avoidance route planning,” presented at the Abstract and Applied Analysis, 2014, vol. 2014.

[7] S. Xie, X. Chu, M. Zheng, and C. Liu, “A composite learning method for multi-ship collision avoidance based on reinforcement learning and inverse control,” *Neurocomputing*, vol. 411, pp. 375–392, 2020.

[8] S. Temizer, M. Kochenderfer, L. Kaelbling, T. Lozano-Pérez, and J. Kuchar, “Collision avoidance for unmanned aircraft using Markov decision processes,” presented at the AIAA guidance, navigation, and control conference, 2010, p. 8040.

[9] H. Y. Ong and M. J. Kochenderfer, “Markov decision process-based distributed conflict resolution for drone air traffic management,” *J. Guid. Control Dyn.*, vol. 40, no. 1, pp. 69–80, 2017.

[10] M. J. Kochenderfer and J. P. Chryssanthacopoulos, “Collision avoidance using partially controlled Markov decision processes,” presented at the Agents and Artificial Intelligence: Third International Conference, ICAART 2011, Rome, Italy, January, 28-30, 2011. Revised Selected Papers 3, 2013, pp. 86–100.

[11] M. Li, H. Bai, and N. Krishnamurthi, “A markov decision process for the interaction between autonomous collision avoidance and delayed pilot commands,” *IFAC-Pap.*, vol. 51, no. 34, pp. 378–383, 2019.

[12] Y. Fu, X. Yu, and Y. Zhang, “Sense and collision avoidance of unmanned aerial vehicles using Markov decision process and flatness approach,” presented at the 2015 IEEE International Conference on Information and Automation, 2015, pp. 714–719.

[13] W. Wang, L. Huang, K. Liu, X. Wu, and J. Wang, “A COLREGs-Compliant Collision Avoidance Decision Approach Based on Deep Reinforcement Learning,” *J. Mar. Sci. Eng.*, vol. 10, no. 7, p. 944, 2022.

[14] X. Ding, S. L. Smith, C. Belta, and D. Rus, “Optimal control of Markov decision processes with linear temporal logic constraints,” *IEEE Trans. Autom. Control*, vol. 59, no. 5, pp. 1244–1257, 2014.

[15] A. U. Udom, “A Markov decision process approach to optimal control of a multi-level hierarchical manpower system,” *CBN J. Appl. Stat.*, vol. 4, no. 2, pp. 31–49, 2013.

[16] T. I. Fossen, *Handbook of marine craft hydrodynamics and motion control*. John Wiley & Sons, 2011.

[17] I. Scholberg, “Modeling and control of underwater vehicle-manipulator systems,” presented at the Proc. Conf. Marine Craft Maneuvering and Control, Southampton, UK, 1994, 1994.

[18] M. Zhu, A. Hahn, Y.-Q. Wen, and W.-Q. Sun, “Optimized support vector regression algorithm-based modeling of ship dynamics,” *Appl. Ocean Res.*, vol. 90, p. 101842, 2019.

[19] Y. Melhaoui, A. Kamil, M. Riouali, K. Mansouri, and M. Rachik, “Optimal control of ship collision avoidance problem,” *J Math Comput Sci*, vol. 12, p. Article-ID, 2022.

[20] M. Abdelaal, M. Fränzle, and A. Hahn, “Nonlinear Model Predictive Control for trajectory tracking and collision avoidance of underactuated vessels with disturbances,” *Ocean Eng.*, vol. 160, pp. 168–180, 2018.

[21] E. L. Porteus, “Stochastic inventory theory,” *Handb. Oper. Res. Manag. Sci.*, vol. 2, pp. 605–652, 1990.

[22] F. Garcia and E. Rachelson, “Markov decision processes,” *Markov Decis. Process. Artif. Intell.*, pp. 1–38, 2013.

[23] H. S. Chang, J. Hu, M. C. Fu, and S. I. Marcus, *Simulation-based algorithms for Markov decision processes*, vol. 68. Springer, 2013.

[24] J. Lisowski, “Computational intelligence methods of a safe ship control,” *Procedia Comput. Sci.*, vol. 35, pp. 634–643, 2014.

[25] X. Yu, X. Zhou, and Y. Zhang, “Collision-free trajectory generation and tracking for UAVs using Markov decision process in a cluttered environment,” *J. Intell. Robot. Syst.*, vol. 93, pp. 17–32, 2019.