symmetry

MDPI

Article

Maintaining Symmetry in Optimal and Safe Control of the Ship to Avoid Collisions at Sea

Józef Lisowski

Faculty of Electrical Engineering, Gdynia Maritime University, 81-225 Gdynia, Poland; j.lisowski@we.umg.edu.pl; Tel.: +48-694-458-333

Abstract: The aim of this study was to make a novel symmetry analysis in relation to the importance of optimizing the ship's trajectory and safety in situations at sea where there is a risk of collision with other ships. To achieve this, the state constraints in the optimization were formulated as ship domains generated by the neural network. In addition, the use of the Bellman dynamic programming method enabled the effective optimization of the ship's safe control. The above assumptions were confirmed by the calculations of the optimal and safe ship traffic paths for the two valid agree with COLREGs states of visibility at sea and for different densities of the dynamic programming grid. Practical conclusions from the research were formulated, and a plan for further research on methods of ensuring safety in navigation was outlined.

Keywords: optimal control; safety control; artificial neural network; computer simulation

Citation: Lisowski, J. Maintaining Symmetry in Optimal and Safe Control of the Ship to Avoid Collisions at Sea. *Symmetry* **2023**, *15*, 1016. https://doi.org/10.3390/sym15051016

Academic Editors: Crescenzo Pepe and Silvia Maria Zanoli

Received: 2 April 2023
Revised: 28 April 2023
Accepted: 30 April 2023
Published: 2 May 2023

1. Introduction

1.1. Related Works

The field of control systems has seen significant progress in recent years with the development of advanced control techniques and technologies. For example, Duan et al. [1] proposed a method to control the formation and constraint of constant-time and time-varying output formation for heterogeneous general multi-agent systems. In the semiconductor industry, Song et al. [2,3] investigated a learning control for motion coordination in wafer scanners, while Li et al. [4] proposed a distributionally reliable method for controlling the predictive stochastic model based on optimization. Moreover, Cao et al. [5] proposed a method to improve the security of the physical layer of the NOMA uplink, using energy harvesting jammers.

Many works concern the use of advanced control techniques in the automation of industrial production processes. Zanoli and Pepe used pressure management of the water distribution system containing a two-layer Model Predictive Control and optimization of pump scheduling in [6], and in [7], a multi-mode Model Predictive Control was proposed for steel billets reheating furnaces control approach together with virtual sensor and control mode selector. Guo J. and Guo H. in [8] presented a real-time risk detection method and protection strategy for intelligent ship network security based on cloud computing.

The following papers were devoted to the symmetry conditions in optimal control. Flaskamp et al. described, in [9], a model predictive control scheme which is based on a library of precomputed motion primitives, illustrated using an academic mobile robot. Danielson and Bornelli demonstrated in [10] that symmetric model predictive control problems produce symmetric controllers, leading to exponential memory reduction and simple, intuitive optimal controllers. Leon et al. showed in [11] that appropriate geometric formulations allow us to reduce the number of equations associated with optimal control problems with symmetry and compare the solutions of the original system with the solutions of the reduced one. Stratoglou et al. [12] examined reduction by symmetry for optimality conditions in optimal control problems of left-invariant affine multi-agent control systems, with partial symmetry breaking cost functions for continuous-time and discrete-time systems. Maidens et al. [13] presented a method of exploiting symmetries of discrete-time optimal control problems to reduce the dimensionality of dynamic programming iterations.

These advances demonstrate the potential of control systems to improve system performance and safety and motivate current research into safe ship control through optimization and dynamic programming.

Many optimization studies are devoted to prevention in shipping. Thus, the use of an evolutionary algorithm for multi-object optimization of the ship's trajectory was presented in [14] by Szlapczynski and Ghaemi. On the other hand, Li proposed in [15] the use of particle swarm and genetic algorithms to determine the ship collision avoidance path. A comparative analysis of safe trajectory planning optimization methods for maritime surface ships was conducted in [16] by Lazarowska. Melhaoui et al. [17] analyzed the optimal control of the ship collision avoidance problem based on Pontryagin's maximum principle. Optimizing the joint collision avoidance operations of multiple ships from an overall perspective was presented in [18] by Li et al.

Examples of optimization of various objects and control processes using dynamic programming were presented in the following papers.

Bai and Zhao [19] and Xie et al. [20] presented the use of approximate dynamic programming for ship course control. The planning of ships and yachts' routes using dynamic programming was presented by Bijlsma [21], Wei and Zhou [22], and Lindberg in [23]. In turn, Esfahani et al. [24] and Geng et al. [25] presented the plans of path autonomous surface ships; Nicholson and Pullen [26] described the use of dynamic programming in ship fleet management; and Zhen et al. [27], Liu et al. [28] and Nguyen [29] discussed the use of heuristic dynamic programming to synthesize optimal decisions regarding ship refueling, mooring control and ship roll stabilization.

The use of dynamic programming in ship hybrid energy management was described by Yuan et al. in [30], and crowd-shipping with in-store customers was discussed by Mousavi et al. in [31].

In addition, the following works are related to the subject of this paper. Bai et al. [32] innovatively quantified the short-to-medium-term operational risk management strategies by using Automatic Identification System (AIS) data. Venturini et al. [33] presented the multi-port berth allocation problem with speed optimization and emission.

Many new articles that dealt with close topics. Yes, Li et al. [34] proposed to improve the signal-to-noise quality of the environmental penetrating radar to restore it and improve the target. Guo and Zhong [35] discussed the pilot city intelligence and evaluated it with a multi-period difference model. Then, Li Q-K. et al. [36] studied the consensus problem for multi-agent supply chain systems in switching topology and uncertain requirements.

Jung and Yoo [37] described the determination of optimal locations for rescue ships by using image processing and clustering, reducing the distance of the rescue vessel to the accident site. Chen et al. [38] performed an analysis of the hydrodynamic interaction of an autonomous floating submersible vehicle and a ship with the effect of waves, which improved the guiding and controlling of AUV movement.

Miller and Walczak [39] proposed the use of Bezler curves, which are a method of predicting trajectories with little computation, to approximate the path of maritime autonomous surface ships (MASSs). Then, the method of detecting and tracking vessels on inland waterways by using enhanced You Look Only Once version 3 (YOLOv3) detection algorithm and Deep Simple Online and DEEP SORT version 3 tracking algorithm were presented by Jie et al. in [40].

Chen et al. [41] proposed the use of the Gaussian regression method to identify dynamic ship properties, achieving greater accuracy in predicting ship movement.

1.2. Problem Statement

Symmetry properties can help in the design and analysis of ship control systems at different levels of their automation hierarchy. A multilevel ship motion control system includes the following hierarchical functional control layers, from lowest to highest: stabilization of course, speed, position and heel compensation; optimal control; adaptive control; and safe control and path optimization.

The task of optimal ship control can be formulated as follows. The course of the steering process of our ship in the group of other ships encountered is described by the equation of state:

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}, t) \tag{1}$$

where $\boldsymbol{x}$ represents state variables, $\boldsymbol{u}$ represents control variables and t is time.

The state and control variables are constrained:

$$\boldsymbol{g}(\boldsymbol{x}, \boldsymbol{u}, t) \geq 0 \tag{2}$$

The control objective function, $\boldsymbol{Q}$, as an index of the optimal ship path has the following form:

$$\boldsymbol{Q} = \boldsymbol{Q}(\boldsymbol{x}, \boldsymbol{u}, t) \rightarrow \min \tag{3}$$

Particularly important is symmetry in terms of the simultaneous fulfillment of the conditions for controlling the optimal and safe movement of the ship in situations where there is a risk of a collision with other ships (see Figure 1).

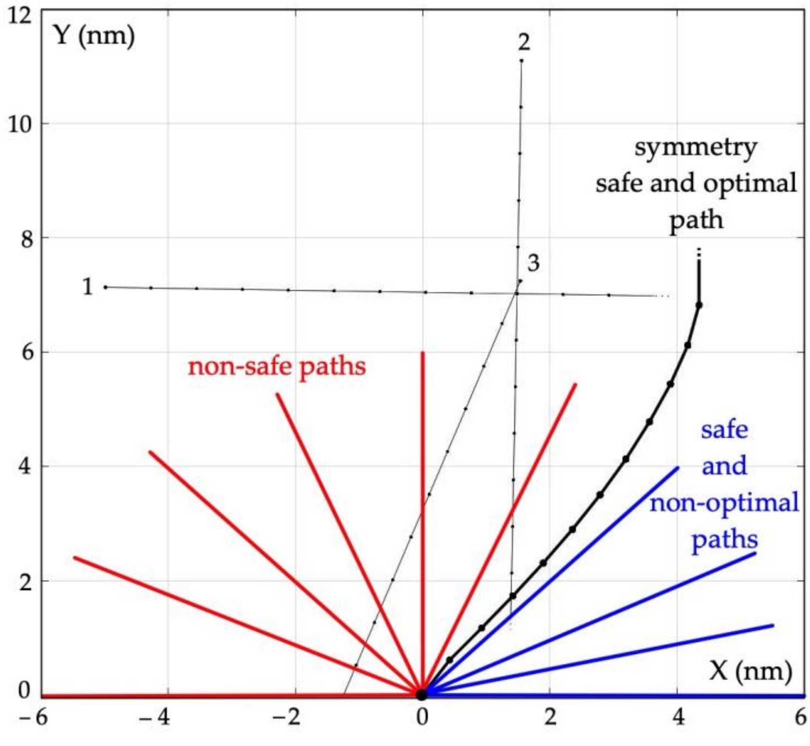

Figure 1. Illustration of symmetry of safe and optimal ship paths in collision situations.

1.3. Contribution

Currently, there is little work on the synthesis and testing of new methods of safe and optimal ship paths in situations at sea where multiple vessels are encountered.

The process of safe ship control can be presented in the form of the following mathematical models:

- A static model, based on the speed triangle, used to determine a safe maneuver to change the ship's course and/or speed;
- Kinematic model, based on the area of permissible maneuvers, allowing us to determine the safe kinematic trajectory of the ship;

- A dynamic model using equations of state dependent on the mathematical description of the ship's hydrodynamics, enabling the determination of the ship's dynamic safe trajectory;
- The game model as a matrix game of many participants that is the basis for determining the game safe trajectory of the ship.

The model presented in this paper takes into account the ship's dynamics, and the Bellman dynamic programming method used for it allows us to determine the optimal safe trajectory, while the share of other ships is mapped in the form of domains generated by an artificial neural network.

Therefore, the proposed safe ship control method is an innovative step to previously published methods.

Contrary to the proposal of Sawada et al. in [42], calculating the obstacle zone by targets and using a bumper model with constant dimensions, this paper proposes a simultaneous solution to the problem of the ship's safe path and the problem of its optimality, based on the generation of kinematic domains of ships of variable size depending on the risk of collision and its multistage dynamic programming.

The novelty of this article is the synthesis and analysis of the symmetry of the safe and optimal ship path in the vicinity of other ships, taking into account both the subjectivity of the navigator in the situation and the limitations of dealing with the need to follow the COLREGs maneuvering rules (Figure 2).

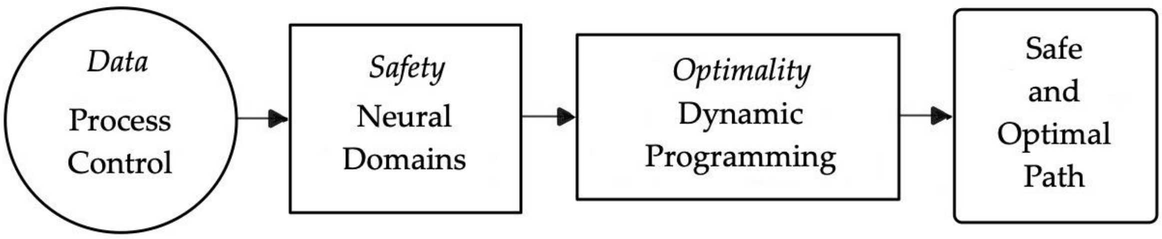

Figure 2. Stages leading to the determination of a safe and optimal ship path.

The scientific goal is to synthesize a new ship path calculation algorithm when passing many other ships. The aim of this research was the experimental analysis of simulation tests on safe and optimal ship control, assuming different values of safe passing distance and different grid densities.

1.4. Work Content

The content of this article is as follows. First, the progress in the field of symmetry in the engineering of control systems for various mobile, autonomous and non-autonomous objects in recent years is described, and the scientific goal of research in the field of symmetry in safe and optimal ship tracks is formulated. Then, the method of safe control using an artificial neural network is presented, and in the next part, the solution to the optimization task, using Bellman's dynamic programming, is presented. The next part contains a description of the algorithm for determining the safe and optimal ship path. Then, the results of the computer simulation of the algorithm in an example scenario of a real navigational situation are presented. After that, some important and valuable issues arising from the conducted research are presented. Finally, the implementation of the thesis formulated in the introduction is confirmed, and directions for future improvements of the subject of research are presented.

2. Safety

To ensure safe navigation while respecting the COLREGs, an appropriate form of state variable constraints is formulated. In good sea visibility, objects approaching from the left are assigned a circle with a radius of safe approach distance, D_s. This also applies to stationary objects.

In the event of an encounter with a vessel approaching from the right, which must be cleared, constraints are assigned in the form of a hexagon, whose dimensions are calculated by an artificial neural network (see Figure 3).

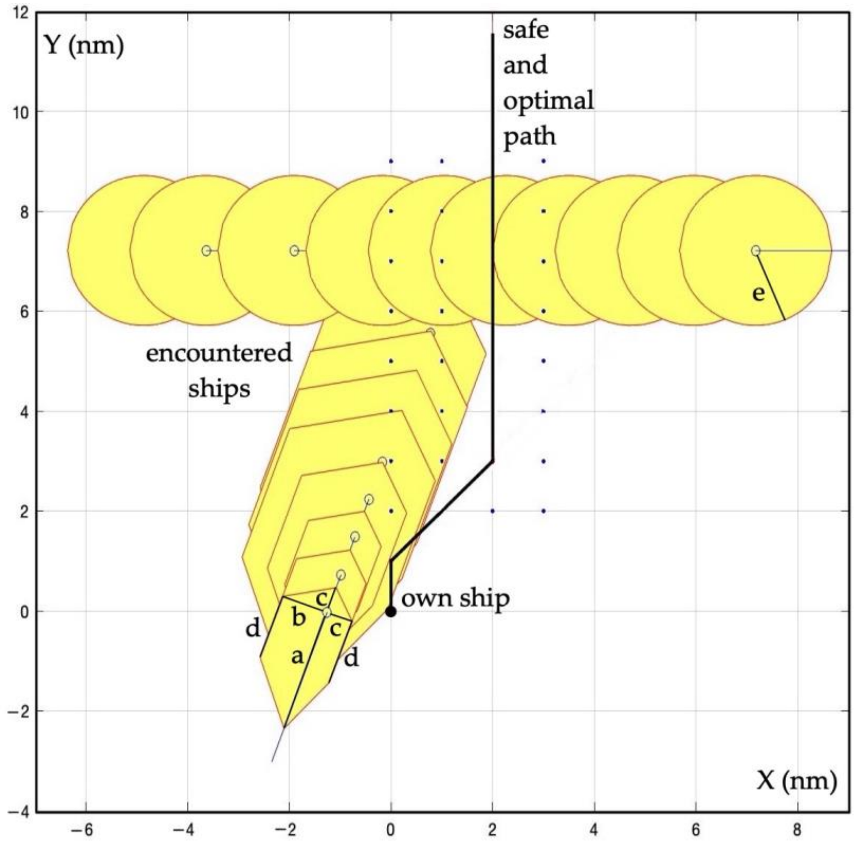

Figure 3. Construction of the shape and dimensions of ship domains in the form of a hexagon and a circle.

The following dimensions of the hexagonal and circle domains are assumed:

$$a = LV_j^{1.26} + 30V_j \quad (4)$$

$$b = BV_j^{0.44} > D_s \quad (5)$$

$$c = 0.5D_s > 0.5\text{ nm} \quad (6)$$

$$d = T_s V_j \quad (7)$$

$$e = D_s \quad (8)$$

where L and B represent the own ship's length and breadth; D_s and T_s are the distance and time to safe approach; and V_j is the encountered j ship velocity [43].

A two-stage shaping of domain sizes during ship traffic is introduced.

- First, the collision risk is reflected in the length of the ship's domain, which depends both on the speed of the encountered vessel (V_j) and on the time (T_s) remaining to reach the safe distance (D_s), according to (4) ÷ (8) equations.
- Then, in the domains' dimensions, navigator subjectivity in collision risk assessment is reflected, using a properly trained artificial neural network.

The neural network acts as a classifier of the degree of risk of collision with passing ships. The designed classifier is a three-layer unidirectional artificial neural network with six neurons in the input layer, three neurons in the hidden layer and one neuron in the output layer (see Figure 4).

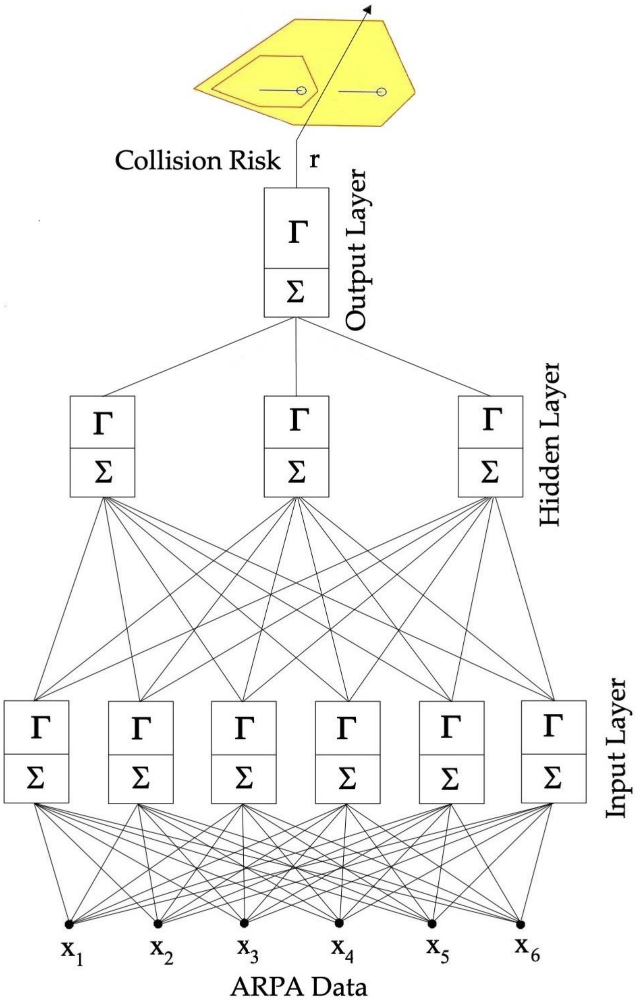

Figure 4. Artificial neural network shaping the size of domains.

Neurons in the input and hidden layers use a sigmoidal bipolar activation function (tangent hyperbolic):

$$f(x) = \frac{1 - e^{-x}}{1 + e^{-x}} \tag{9}$$

Meanwhile, the neuron of the output layer has a sigmoidal unipolar activation function:

$$f(x) = \frac{1}{1 + e^{-x}} \tag{10}$$

In simple terms, the operation of the presented artificial neural network can be presented as follows:

$$r = \Gamma\,[Wx] \tag{11}$$

where Γ represents activation functions of individual neurons, W is the matrix of weights and x is an input data vector fed to the first layer of neurons.

The input data vector is defined as follows:

$$x = [x_1, x_2, x_3, x_4, x_5, x_6] = \left[D_j, N_j, V_j, \psi_j, V, \psi\right] \tag{12}$$

where D_j is distance to the j-th ship; N_j is bearing to the j-th ship; V_j is the j-th ship speed; ψ_j is the j-th ship course; V is own ship's speed; and ψ is the own ship's course.

Meanwhile, r is the response of the network to the input data vector, the value of which is between 0 and 1, representing the collision risk:

$$r = \left\{ \begin{array}{l} 0.1 \rightarrow \text{safe situation} \\ 0.3 \rightarrow \text{attention} \\ 0.5 \rightarrow \text{collision risk} \\ 0.7 \rightarrow \text{dangerous situation} \\ 0.9 \rightarrow \text{collision} \end{array} \right\} \tag{13}$$

Then, the network response is used to tune the size of the domains according to the collision risk value, as follows:

$$NetResp = logsig(netsum(nw3 * tansig(netsum(nw2 * tansig(netsum((nw1 * p), nb1)), nb2)), nb3)) \tag{14}$$

where *logsig* is unipolar sigmoid activation function of output neuron activation; *tansig* is tangent hyperbolic agtication function of hidden layer neurons; *netsum* is weighted sum of signals coming to each neuron; $nw1,2,3$ are weights of individual connections; $nb1,2,3$ are braking coefficients.

We then have the following:

$$NetResp = 0.5 + NetResp \tag{15}$$

The dependence (12) was adopted due to the limitation of the unfavorable impact of small network response values on the dimensions of ship domains, which should be non-zero even for safe situations.

To change the size of domains in the form of hexagons, matrixes of values (a,b,c,d) are created describing their sizes, which are modified by the response of the network estimating the risk of collision:

$$(a,b,c,d) = (a,b,c,d)\ NetResp \tag{16}$$

For restrictions in the form of circles occurring in poor visibility, the radius of the circle is changed, the initial value of which is equal to the safe passing distance, D_s:

$$e = D_s\ NetResp \tag{17}$$

Domain aspect ratio values are limited in the following range:

$$\begin{array}{c} 1.2 < a < 4 \\ D_s < b < 3.5 \\ 0.5 < c < 1.2D_s \\ 1 < d < 2 \\ 1 < e < 4 \end{array} \tag{18}$$

Using MATLAB version R2023a Neural Network Toolbox software, the artificial neural network was synthesized and then trained according to the error backpropagation algorithm with adaptive learning rate and momentum.

The learning material consisted of navigational scenarios. About 300 responses were recorded from experienced navigators in ARPA system training courses.

For each scenario, navigators assessed collision risk and subjectively selected, in accordance with good sea practice, the best anticollision maneuver to change the ship's course or speed. Therefore, the neural network trained in this way represents the average experience of a large marine navigator group. The size of the ship domain depends on the collision risk value being neural network output.

3. Optimality

Among the many dynamic optimization methods, Bellman dynamic programming is best suited for the synthesis of optimal ship motion control [44–47].

Dynamic programming is defined as a computer programming technique where an algorithmic problem is first broken down into subproblems; the results are saved; and then subproblems are optimized to find an overall solution, which usually has to do with finding the maximum and minimum range of the algorithmic problem.

Some of the primary dynamic programming algorithms in use are the greedy, Floyd–Warshall and Bellman–Ford algorithms.

The following groups of dynamic programming applications can be distinguished.

- Determining the number of ways to cover a distance;
- Identifying the optimal strategy of a game;
- Takeaways.

In addition, this author’s articles [48,49] present optimization methods for use in maritime transport and logistics.

This paper proposes the following use of neural state constraints in dynamic programming to calculate an optimal and safe ship path in a collision situation with other ships.

Figure 5 illustrates a dynamic programming grid of K steps and N nodes, with the final conditions in the form of the final course and the final point of the cruise path.

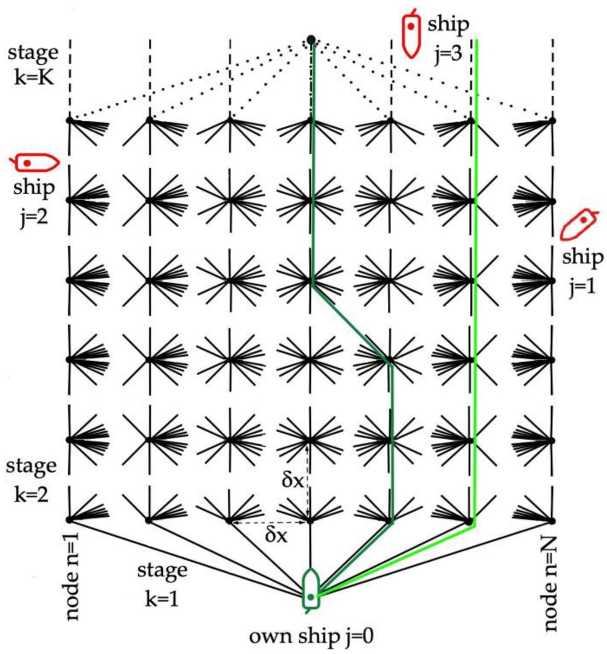

Figure 5. Dynamic programming grid consisting of K stages and N nodes with end conditions in the form of final course and final point of the cruise path: $k = 1, 2, \ldots, K$ is number of stages; $n = 1, 2, \ldots, N$ is number of nodes; $j = 0, 1, 2, \ldots, J$ is number of ships; δx is the distance between nodes, which determines the density of the grid.

First, starting from the first step, for each path between the nodes, the time of its traversal is calculated.

If the encountered ship passes through a given node, then instead of the previously calculated passage time, a much longer time is entered as a penalty.

Before moving on to the next node, only one of the calculated paths with the shortest travel time is remembered.

The last stage allows for the implementation of the final conditions of optimization of the entire route of the ship, with data either in the form of the final course or final point. In the last stage, the path with the shortest passage time of the entire ship is selected as the optimal one.

In order to determine a safe and optimal ship path, it moves end-to-end through the nodes, selecting those paths that were previously remembered with the shortest traversal time.

4. Algorithm 1: Safe and Optimal Path

Combining the previously presented procedures of the artificial-neural-network-generated time-varying domains of passing ships reflecting traffic safety and dynamic programming allowing for the calculation of an optimal ship path, an SOP algorithm for determining the Safe and Optimal Path in a collision situation was obtained. The SOP algorithm was written and tested in MATLAB/Simulink version R2023a software (Agorithm 1).

Algorithm 1: Safe and Optimal Path

```
Begin
Input and development of initial data
Set K = 1;
K = number of stages in the simulation
Calculation of X1, X2, dt
Calculation of domains
Check if calculations meet domains
if V1 = 0.0001 Then
    Manewr Crash
    Return
    Else
    Set K = 2;
    j = number of nodes in the simulation
    For j = 0
        Calculation of X1, X2, dt
        Time = Time + dt;
        Calculation of domains
        Check if calculations meet domains
        Save the best Time
        j−−;
        V = V − dV;
    End
    Set K = 3;
    For K = number of stages
        j = number of nodes in the simulation;
        For j = 0
            i = number of nodes in the simulation
            For i = 0
                Calculation of X1, X2, dt
                Time = Time + dt;
                Calculation of domains
                Check if calculations meet domains
                Save the best Time
                V = V − dV;
                i−−;
            End
            j−−;
        End
        K++;
    End
    Print Calculations of safe and optimal ship path
    Visualize navigational situation
End
```

5. Computer Simulation Results

The iMac 21.5″ personal computer with the following equipment was used for the simulation:

- 3.6 GHz CPU, Quad Core Intel Core i7;
- Graphic Radeon Pro 560 4 GB;
- RAM 32 GB, 2.4 GHz, DDR4;
- Mac OS Ventura version 13.3.1 software;
- Macintosh HD 1 TB storage;
- Retina display 4096 × 2304.

The algorithm was tested in an example scenario of a real navigational situation recorded in the Baltic Sea by the onboard ARPA anticollision system on the r/v Horyzont II research and training vessel (Table 1 and Figure 6).

Table 1. Data of ships in navigation situation in the Baltic Sea.

Ship j	Speed V_j (kN)	Course ψ_j (°)	Distance D_j (nm)	Bearing N_j (°)
0	15.0	0	0	0
1	14.4	91	8.7	325
2	16.3	181	11.2	8
3	16.0	201	7.4	12
4	15.0	85	6.0	300
5	0	0	−3.0	280

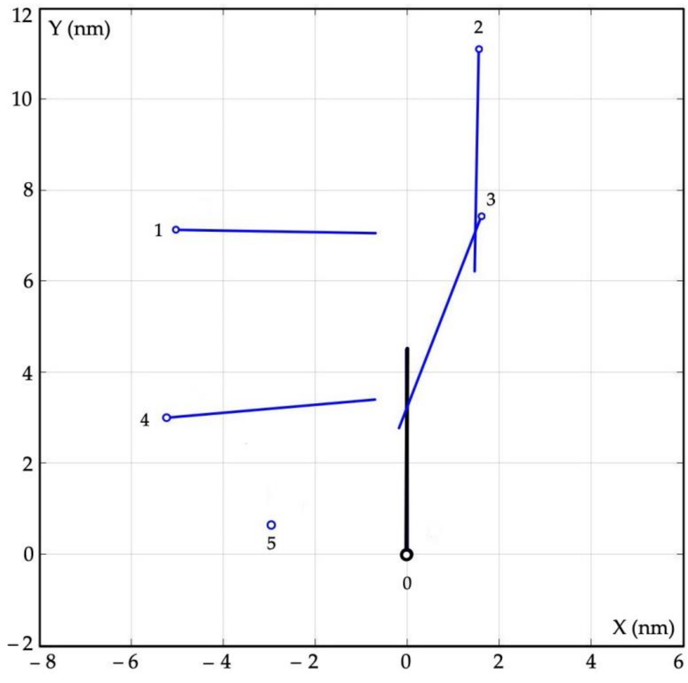

Figure 6. Movement situation showing own ship and five other encountered ships.

The situation shows the movement of one's own ship with speed (V) and heading (ψ) while it passes other ships (j), which are located at distances (D_j) and bearings (N_j) and moving at speeds (V_j) and headings (ψ_j).

First, the algorithm was tested in good and restricted visibility, with different values of grid density (δx) and safety distance (D_s).

A comparison of safe and optimal ship paths for two different grid densities with good visibility at sea for D_s = 0.1 nm is shown in Figure 7, and one for for D_s = 1.0 nm is shown in Figure 8.

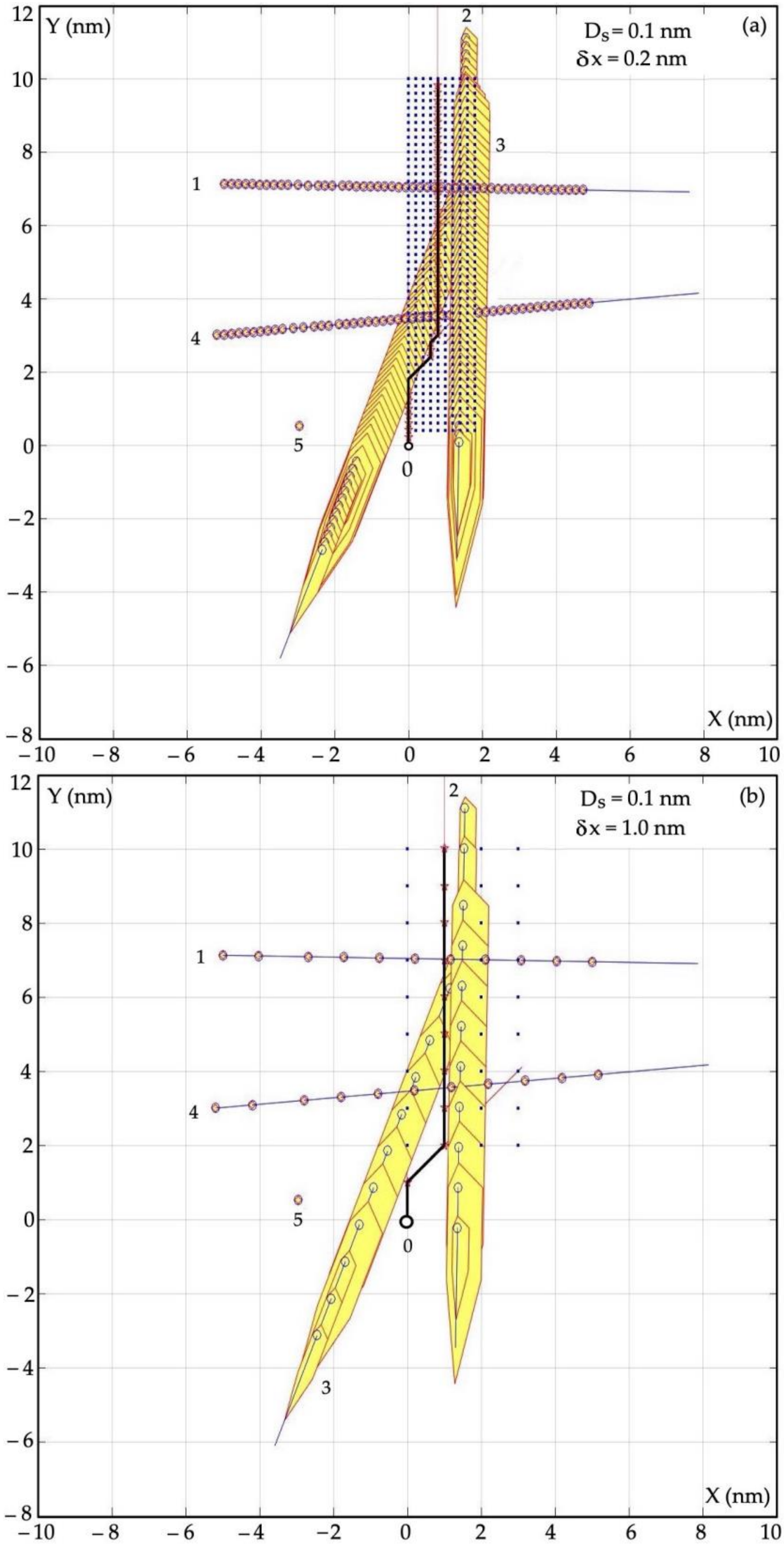

Figure 7. Safe and optimal own ship path in good sea visibility at D_s = 0.1 nm and grid density: (**a**) δx = 0.2 nm and (**b**) δx = 1.0 nm.

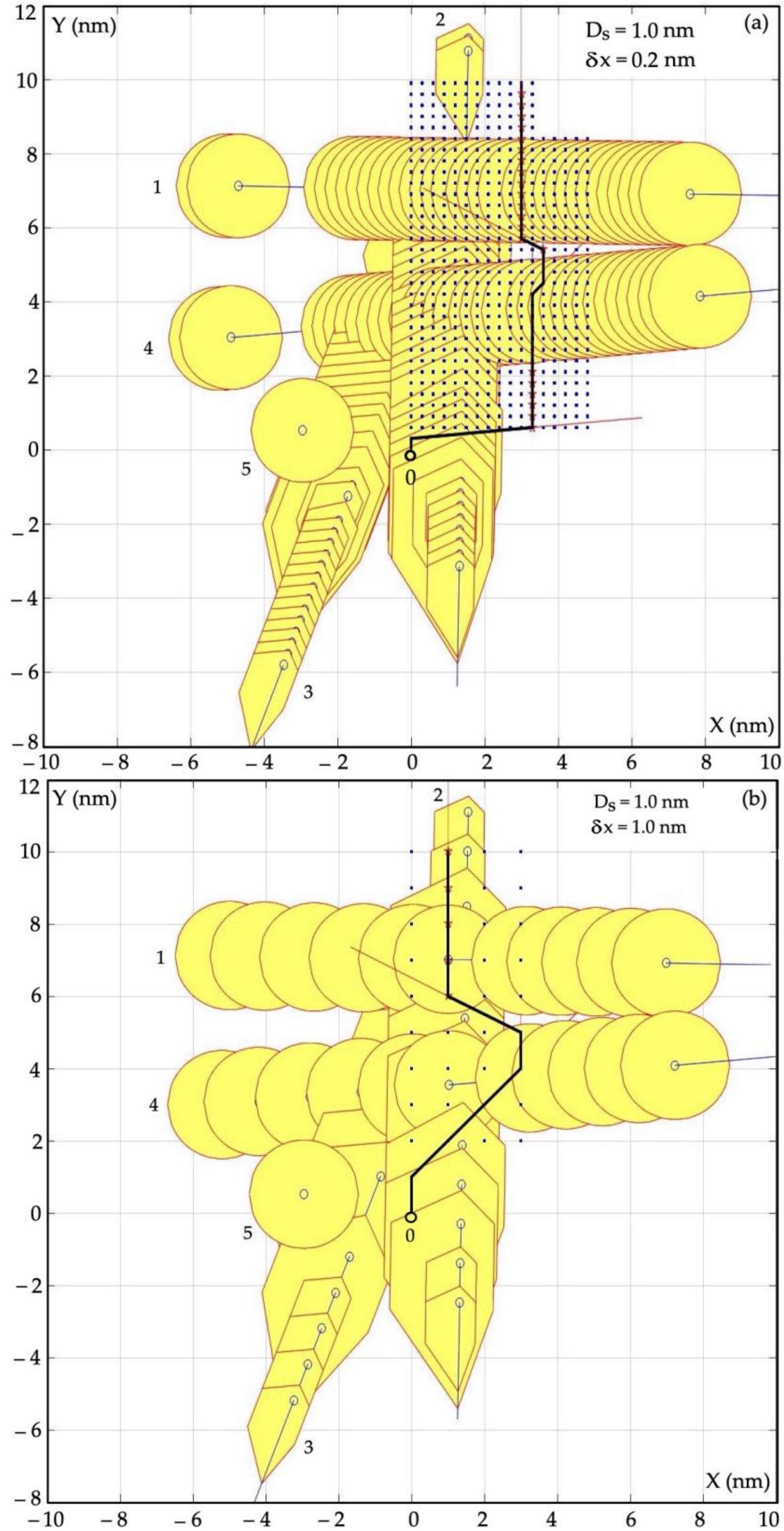

Figure 8. Safe and optimal own ship path in good sea visibility at D_s = 1.0 nm and grid density: (**a**) δx = 0.2 nm and (**b**) δx = 1.0 nm.

Figure 9 illustrates the dependence of the minimum time of a safe and optimal ship path in good sea visibility on the value of safe distance (D_s) and grid density (δx).

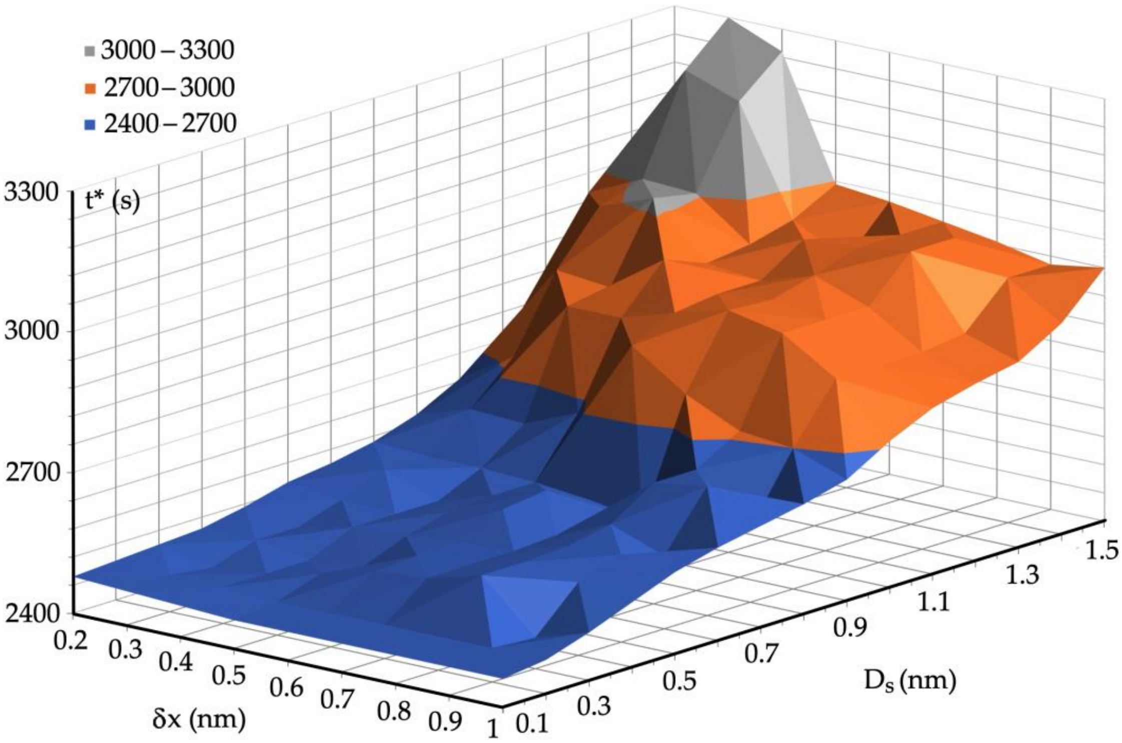

Figure 9. Dependence of the minimum safe and optimal ship path time, t^*, with good visibility at sea on the safe passage distance (D_s) and the grid density (δx).

The sensitivity function, s_{t^*}, of the time-optimal ship control can be represented as a partial derivative of the minimum execution time path, t^*, with respect to changes in the p parameters [50,51]:

$$s_{t^*} = \frac{\partial t^*}{\partial p} = \frac{t^*(p_0) - t^*(p)}{t^*(p_0)} 100\% \quad (19)$$

where $p(\delta x, D_s)$ represents variable parameters in the form of grid density (δx) and safe passing distance (D_s), and $p_0(\delta x_0, D_{s0})$ is a fixed reference point of these parameters (Table 2).

Table 2. Sensitivity s_{t^*} (%) of optimal control to changes in safe passage distance, D_s (nm), and grid density, δx (nm), in conditions of good visibility at sea.

δx \ D_s	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0	1.1	1.2	1.3	1.4	1.5
0.2	3.8	3.6	3.4	3.2	2.6	1.8	1.4	1.2	0.2	1.8	3.4	7.8	12.2	14.6	14.4
0.3	4.0	3.8	4.2	3.2	3.0	2.2	1.0	1.0	1.4	4.2	9.4	13.0	14.0	15.6	15.8
0.4	4.0	3.8	3.8	3.8	2.4	2.4	2.4	1.0	1.2	8.6	9.0	14.2	13.8	14.0	14.2
0.5	4.2	4.0	3.0	2.8	2.2	1.8	1.2	1.2	2.8	8.6	11.8	10.6	11.0	13.8	13.4
0.6	4.2	4.0	2.0	1.8	1.8	0.4	0	1.6	8.4	9.2	10.2	10.2	12.2	9.2	13.0
0.7	4.4	3.2	0.8	1.6	1.6	1.4	1.4	2.2	6.8	10.8	10.6	11.4	12.4	12.4	12.4
0.8	4.2	4.0	0.3	0.6	0.4	0.2	2.2	2.2	4.0	10.0	10.0	10.0	13.0	13.0	11.8
0.9	4.2	4.0	1.5	1.2	0.2	1.2	2.8	1.4	3.6	4.8	7.6	7.6	8.0	11.0	11.0
1.0	4.2	4.0	1.5	1.8	0.6	0.2	0.8	1.4	2.2	4	4.9	6.2	6.8	4.3	11.6

Optimal control sensitivity, not exceeding 5% (see green color in Table 2) in conditions of good visibility at sea, is obtained at the value of the safe passage distance in the range of 0.1 nm to 0.8 nm and for the grid density from 0.2 nm to 1.0 nm.

A comparison of safe and optimal ship paths for two different grid densities with restricted visibility at sea for D_s = 1.0 nm is shown in Figure 10, and a comparison for D_s = 3.0 nm is shown in Figure 11.

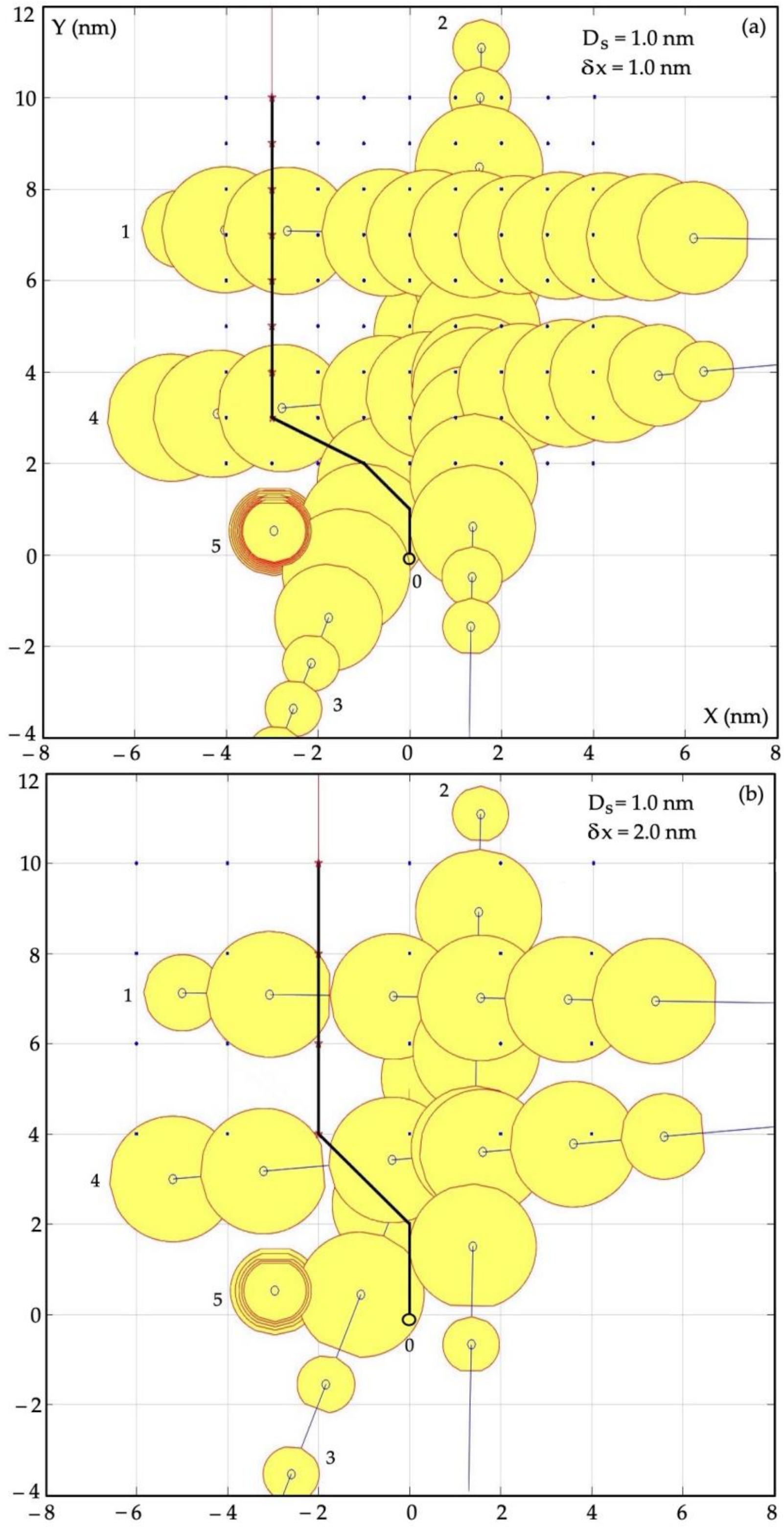

Figure 10. Safe and optimal own ship path in restricted sea visibility at D_s = 1.0 nm and grid density: (**a**) δx = 1.0 nm and (**b**) δx = 2.0 nm.

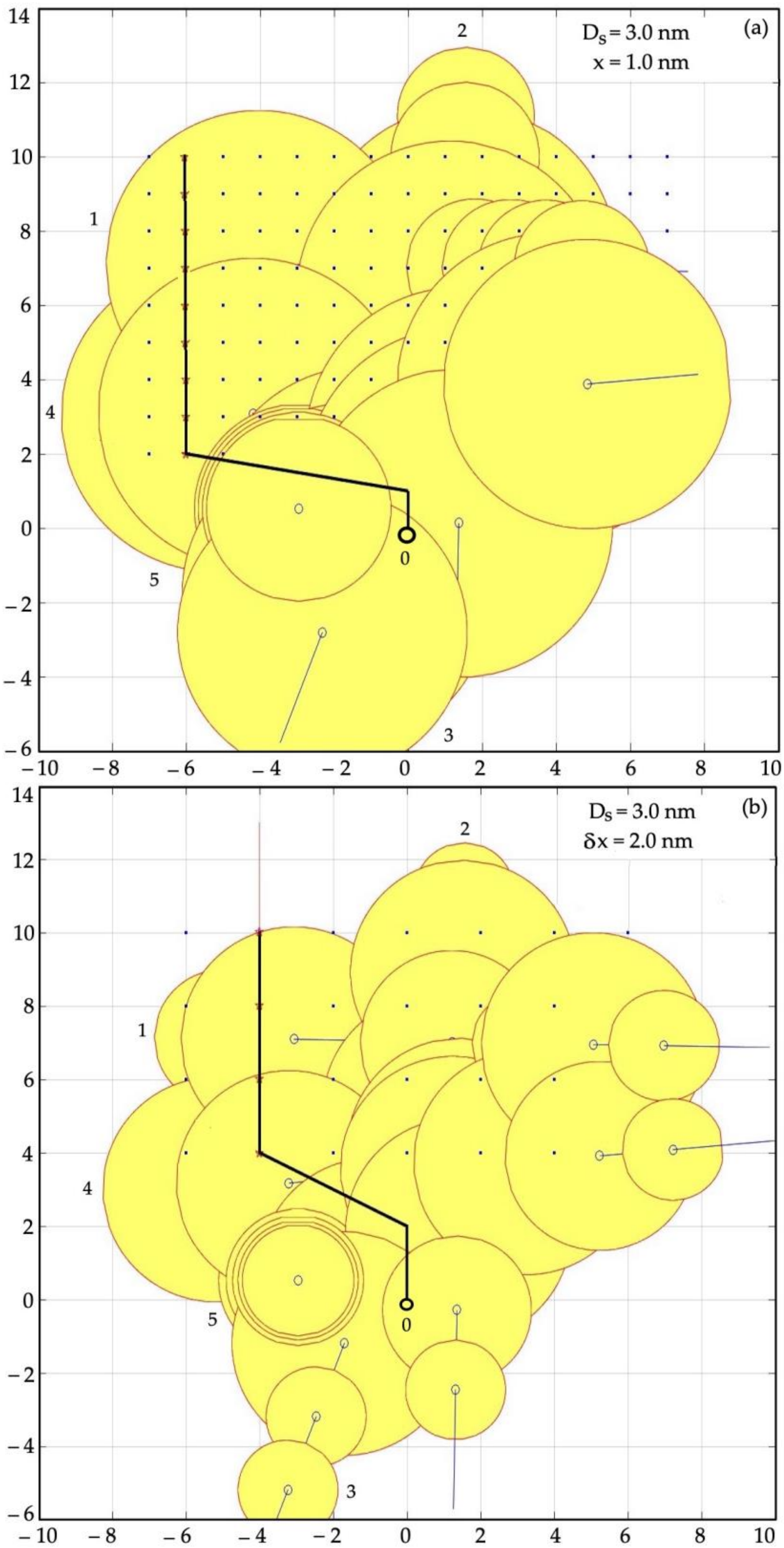

Figure 11. Safe and optimal own ship path in restricted sea visibility at D_s = 3.0 nm and grid density: (**a**) δx = 1.0 nm and (**b**) δx = 2.0 nm.

Figure 12 shows the dependence of the minimum safe and optimal ship path time in conditions of restricted sea visibility on the value of the safe distance (D_s) and the grid density (δx), and then Table 3 shows the value of optimal control sensitivity.

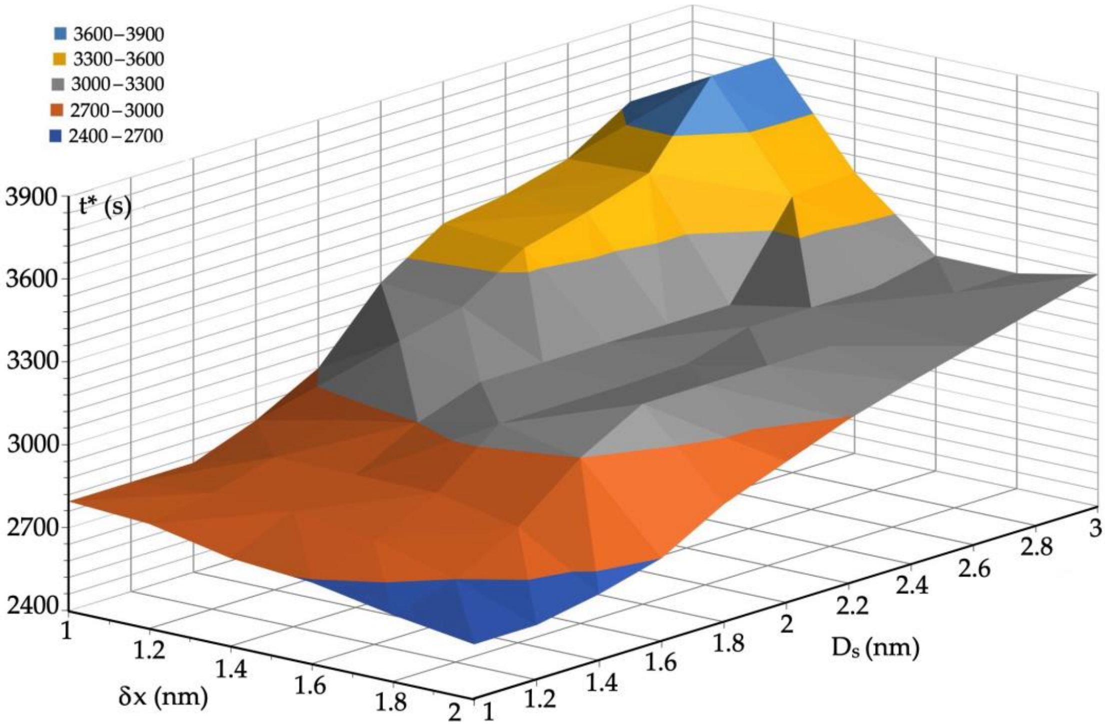

Figure 12. Dependence of the minimum execution time of safe and optimal own ship path with restricted visibility at sea on the safe passing distance, D_s, and the grid density, δx.

Table 3. Sensitivity, s_{t^*} (%), of optimal control to changes in safe passage distance, D_s (nm), and grid density, δx (nm), in conditions of restricted visibility at sea.

D_s / δx	1.0	1.2	1.4	1.6	1.8	2.0	2.2	2.4	2.6	2.8	3.0
1.0	9.2	8.6	8.0	6.6	7.0	7.4	10.2	11.4	13.4	16.0	19.4
1.2	10.0	8.6	5.0	4.4	2.8	4.6	7.0	11.4	12.8	14.0	14.4
1.4	12.0	8.6	5.0	0	0	0	0	0	0	10.8	9.4
1.6	12.8	10.0	4.9	0.6	0.6	0.6	2.4	1.8	1.8	2.6	4.0
1.8	14.6	12.6	8.6	2.6	1.6	1.6	1.6	1.6	1.6	3.4	3.4
2.0	19.8	16.0	14.6	12.6	4.9	3.6	2.8	1.0	2.6	3.4	3.8

Optimal control sensitivity, not exceeding 5% (see green color in Table 3) in conditions of restricted visibility at sea, is obtained at the value of the safe passage distance in the range of 1.4 nm to 3.0 nm and for the grid density from 1.2 nm to 2.0 nm.

6. Discussion

Increasing the density of the grid (δx) dynamic programming leads to a smoother path shape but greater path deviation from the initial direction. On other hand, increasing the safe distance, D_s, lengthens the path and leads to an increase in final path deviation from its initial direction.

Depending on the number of nodes of the dynamic programming grid, which, in the simulations, ranged from 40 to 1000, the CPU running time ranged from 800 ms to 2800 ms.

When selecting the value of grid density, δx, its reference to the value of the safe passing distance, D_s, that is, $D_s/\delta x$, should be considered; it should be between 1 and 2 in order to obtain the correct calculation.

In practice, in accordance with the COLREGs, the value of the safe passing distance assumed in good sea visibility is D_s = 0.2 ÷ 1 nm, and for restricted visibility, it is D_s = 1 ÷ 3 nm. Therefore, calculations, simulations and the subsequent analysis of their results should be carried out separately for these two conditions of sea visibility.

7. Conclusions

The algorithms presented in the article and their simulation tests confirm the thesis adopted at the beginning that it is possible to maintain the symmetry of ship control in collision situations by using the dynamic programming method to optimize it and assess the collision risk neural network in order to ensure the safety of ship traffic in accordance with the rules of COLREGs.

Compared to previous studies in this field, safety and optimality of collision avoidance were similarly assessed separately, but no symmetry analysis was performed between the two requirements.

The formulation of the final optimization conditions made it possible to refer to the actual sailing conditions of the vessel both in the closed and open seas.

The analysis of the sensitivity of the optimal and safe steering in collision situations allowed for the selection of the best grid density values for the dynamic programming of the ship's track and determined better conditions for the selection of the safe passing distance value for both good and limited visibility at sea, in accordance with the COLREG requirements.

In future research, further refinement of the task of maintaining safe and optimal control symmetry should be considered for both non-autonomous and autonomous surface ships, including the following:

- Improving the accuracy of the implementation of a safe and optimal path by introducing an appropriate non-linear dynamic ship model;
- Analysis of other possible optimization criteria and selection of the most adequate one;
- Development of game-acting ship models that would take into account non-compliance with COLREG rules leading to accidental ship collisions;
- Synthesis of ship control game algorithms;
- Increasing the accuracy of navigation information sources by testing the sensitivity of safe and optimal ship control to the inaccuracy of measuring devices.

Funding: This research was funded by a research project of the Electrical Engineering Faculty, Gdynia Maritime University, Poland, No. WE/2023/PZ/02: "Control algorithms synthesis of autonomous objects".

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The study did not report any data.

Conflicts of Interest: The author declares no conflict of interest regarding the publication of this paper. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

References

1. Duan, J.; Duan, G.; Cheng, S.; Cao, S.; Wang, G. Fixed-time time-varying output formation–containment control of heterogeneous general multi-agent systems. *ISA Trans.* **2023**, *1*, 1–20. [CrossRef] [PubMed]
2. Song, F.; Liu, Y.; Shen, D.; Li, L.; Tan, J. Learning Control for Motion Coordination in Wafer Scanners: Toward Gain Adaptation. *IEEE Trans. Ind. Electron.* **2022**, *69*, 13428–13438. [CrossRef]
3. Song, F.; Liu, Y.; Jin, W.; Tan, J.; He, W. Data-Driven Feedforward Learning With Force Ripple Compensation for Wafer Stages: A Variable-Gain Robust Approach. *IEEE Trans. Neural Netw. Learn. Syst.* **2022**, *33*, 1594–1608. [CrossRef]
4. Li, B.; Tan, Y.; Wu, A.; Duan, G. A distributionally robust optimization based method for stochastic model predictive control. *IEEE Trans. Autom. Control* **2021**, *67*, 5762–5776. [CrossRef]

5. Cao, K.; Wang, B.; Ding, H.; Lv, L.; Dong, R.; Cheng, T.; Gong, F. Improving Physical Layer Security of Uplink NOMA via Energy Harvesting Jammers. *IEEE Trans. Inf. Forensics Secur.* **2021**, *16*, 786–799. [CrossRef]
6. Zanoli, S.M.; Pepe, C.; Astolfi, G.; Orlietti, L. Applications of Advanced Process Control Techniques to an Italian Water Distribution Network. *IEEE Trans. Control Netw. Syst.* **2022**, *9*, 1767–1779. [CrossRef]
7. Zanoli, S.M.; Pepe, C.; Orlietti, L. Multi-Mode Model Predictive Control Approach for Steel Billets Reheating Furnaces. *Sensors* **2023**, *23*, 3966. [CrossRef]
8. Guo, J.; Guo, H. Real-Time Risk Detection Method and Protection Strategy for Intelligent Ship Network Security Based on Cloud Computing. *Symmetry* **2023**, *15*, 988. [CrossRef]
9. Flaßkamp, K.; Ober-Blöbaum, S.; Peitz, S. Symmetry in Optimal Control: A Multiobjective Model Predictive Control Approach. In *Advances in Dynamics, Optimization and Computation*; Junge, O., Schütze, O., Froyland, G., Ober-Blöbaum, S., Padberg-Gehle, K., Eds.; SON 2020; Studies in Systems, Decision and Control; Springer: Cham, Switzerland, 2020; Volume 304. [CrossRef]
10. Danielson, C.R.; Borrelli, F. Symmetric Constrained Optimal Control. *IFAC-PapersOnLine* **2015**, *48*, 366–371. [CrossRef]
11. León, M.; Cortés, J.; Diego, D.M.; Martínez, S. General symmetries in optimal control. *Rep. Math. Phys.* **2004**, *53*, 55–78. [CrossRef]
12. Stratoglou, E.; Simoes, A.A.; Colombo, L.J. Reduction in optimal control with broken symmetry for collision and obstacle avoidance of multi-agent system on Lie groups. *Optim. Control* **2022**, *1*, 1–20. [CrossRef]
13. Maidens, J.; Barrau, A.; Bonnabel, S.; Arcak, M. Symmetry reduction for dynamic programming. *Automatica* **2018**, *97*, 367–375. [CrossRef]
14. Szlapczynski, R.; Ghaemi, H. Framework of an evolutionary multi-objective optimisation method for planning a safe trajectory for a marine autonomous surface ship. *Pol. Marit. Res.* **2019**, *26*, 69–79. [CrossRef]
15. Li, N. Research on Ship Collision Avoidance Path Optimization Based on Particle Swarm Optimization and Genetic Algorithm. *Am. J. Math. Comput. Model* **2021**, *6*, 81–87. [CrossRef]
16. Lazarowska, A. *Safe Trajectory Planning for Maritime Surface Ships*; Springer: Berlin/Heidelberg, Germany, 2022; Volume 13, pp. 1–185. [CrossRef]
17. Melhaoui, Y.; Kamil, A.; Riouali, M.; Mansouri, K.; Rachik, M. Optimal control of ship collision avoidance problem. *J. Math. Comput. Sci.* **2022**, *12*, 1–19. [CrossRef]
18. Li, S.; Liu, J.; Negenborn, R.R.; Ma, F. Optimizing the joint collision avoidance operations of multiple ships from an overall perspective. *Ocean. Eng.* **2019**, *191*, 106511. [CrossRef]
19. Bai, X.; Yi, J.; Zhao, D. Approximate Dynamic Programming for Ship Course Control. In *Advances in Neural Networks, Lecture Notes in Computer Science*; Liu, D., Fei, S., Hou, Z.G., Zhang, H., Sun, C., Eds.; Springer: Berlin/Heidelberg, Germany, 2007; Volume 4491. [CrossRef]
20. Xie, Q.; Luo, B.; Tan, F. Optimal tracking control for ship course using approximate dynamic programming method. In Proceedings of the 32nd Chinese Control Conference, Xi'an, China, 26–28 July 2013; pp. 2911–2916, ISBN 978-9-8815-6383-5.
21. Bijlsma, S.J. On the Applications of Optimal Control Theory and Dynamic Programming in Ship Routing. *Navigation* **2002**, *49*, 71–80. [CrossRef]
22. Wei, S.; Zhou, P. Development of a 3D Dynamic Programming Method for Weather Routing. *TRANSNAV Int. J. Mar. Navig. Saf. Sea Transp.* **2012**, *6*, 79–85.
23. Lindberg, M. *Optimizing Yacht Routes using Dynamic Programming*; Lund Institute of Technology: Lund, Sweden, 2004; pp. 1–36, ISSN 0280-5316.
24. Esfahani, H.N.; Szłapczyński, R. Robust-adaptive dynamic programming-based time-delay control of autonomous ships under stochastic disturbances using an actor-critic learning algorithm. *J. Mar. Sci. Technol.* **2021**, *26*, 1262–1279. [CrossRef]
25. Geng, X.; Wang, Y.; Wang, P.; Zhang, B. Motion Plan of Maritime Autonomous Surface Ships by Dynamic Programming for Collision Avoidance and Speed Optimization. *Sensors* **2018**, *19*, 434. [CrossRef]
26. Nicholson, T.A.J.; Pullen, R.D. Dynamic Programming Applied to Ship Fleet Management. *J. Oper. Res. Soc.* **1971**, *22*, 211–220. [CrossRef]
27. Zhen, L.; Wang, S.; Zhuge, D. Dynamic programming for optimal ship refueling decision. *Transp. Res. Part E Logist. Transp. Rev.* **2017**, *100*, 63–74. [CrossRef]
28. Liu, Q.; Li, T.; Shan, Q.; Yu, R.; Gao, X. Virtual guide automatic berthing control of marine ships based on heuristic dynamic programming iteration method. *Neurocomputing* **2021**, *437*, 289–299. [CrossRef]
29. Nguyen, T.D.; Nguyen, Q.D. Optimal Control for Ship Fin Stabilizer System Based on Action Dependent Heuristic Dynamic Programming. *J. Mech. Eng. Res. Dev.* **2021**, *44*, 102–112.
30. Yuan, Y.; Chen, M.; Wang, J.; Yu, W.; Shen, B. A novel hybrid energy management strategy of a diesel-electric hybrid ship based on dynamic programing and model predictive control. *Proc. Inst. Mech. Eng. Part M J. Eng. Marit. Environ.* **2022**, *236*, 644–657. [CrossRef]
31. Mousavi, K.; Bodur, M.; Cevik, M.; Roorda, M.J. *Approximate Dynamic Programming for Crowd-Shipping with In-Store Customers*; University of Wisconsin-Madison: Madison, WI, USA, 2021; pp. 1–47. Available online: https://optimization-online.org/?p=17537 (accessed on 20 April 2023).
32. Bai, X.; Cheng, L.; Iris, C. Data-driven financial and operational risk management: Empirical evidence from the global tramp shipping industry. *Transp. Res. Part E Logist. Transp. Rev.* **2022**, *158*, 1366–5545. [CrossRef]

33. Venturini, G.; Iris, C.; Kontovas, C.A.; Larsen, A. The multi-port berth allocation problem with speed optimization and emission considerations. *Transp. Res. Part D Transp. Environ.* **2017**, *54*, 142–159. [CrossRef]
34. Li, R.; Zhang, H.; Chen, Z.; Yu, N.; Kong, W.; Li, T.; Wang, E.; Wu, X.; Liu, Y. Denoising method of ground-penetrating radar signal based on independent component analysis with multifractal spectrum. *Measurement* **2022**, *192*, 110886. [CrossRef]
35. Guo, Q.; Zhong, J. The effect of urban innovation performance of smart city construction policies: Evaluate by using a multiple period difference-in-differences model. *Technol. Forecast. Soc. Chang.* **2022**, *184*, 122003. [CrossRef]
36. Li, Q.-K.; Lin, H.; Tan, X.; Du, S. H∞ Consensus for Multiagent-Based Supply Chain Systems Under Switching Topology and Uncertain Demands. *IEEE Trans. Syst. Man Cybern. Syst.* **2020**, *50*, 4905–4918. [CrossRef]
37. Jung, C.-Y.; Yoo, S.-L. Optimal Rescue Ship Locations Using Image Processing and Clustering. *Symmetry* **2019**, *11*, 32. [CrossRef]
38. Chen, C.-W.; Chen, Y.; Cai, Q.-W. Hydrodynamic-Interaction Analysis of an Autonomous Underwater Hovering Vehicle and Ship with Wave Effects. *Symmetry* **2019**, *11*, 1213. [CrossRef]
39. Miller, A.; Walczak, S. Maritime Autonomous Surface Ship's Path Approximation Using Bézier Curves. *Symmetry* **2020**, *12*, 1704. [CrossRef]
40. Jie, Y.; Leonidas, L.; Mumtaz, F.; Ali, M. Ship Detection and Tracking in Inland Waterways Using Improved YOLOv3 and Deep SORT. *Symmetry* **2021**, *13*, 308. [CrossRef]
41. Chen, G.; Wang, W.; Xue, Y. Identification of Ship Dynamics Model Based on Sparse Gaussian Process Regression with Similarity. *Symmetry* **2021**, *13*, 1956. [CrossRef]
42. Sawada, R.; Sato, K.; Majima, T. Automatic ship collision avoidance using deep reinforcement learning with LSTM in continuous action spaces. *J. Mar. Sci. Technol.* **2021**, *26*, 509–524. [CrossRef]
43. Lisowski, J. Game strategies of ship in the collision situations. *TransNav—Int. J. Mar. Navig. Saf. Sea Transp.* **2014**, *8*, 69–77. [CrossRef]
44. Sniedovich, M. *Dynamic Programming: Foundations and Principles*; Taylor & Francis: Milton Park, UK, 2010; ISBN 978-0-8247-4099-3.
45. Lisowski, J. Synthesis of a Path-Planning Algorithm for Autonomous Robots Moving in a Game Environment during Collision Avoidance. *Electronics* **2021**, *10*, 675. [CrossRef]
46. Rahmanian, H.; Warmuth, M.K. Online Dynamic Programming. 2017. Available online: https://arhiv.org/abs/1706.00834v3 (accessed on 22 April 2023).
47. Zhang, H.; Liu, D.; Luo, Y.; Wang, D. *Adaptive Dynamic Programming for Control*; Springer: London, UK, 2013; ISBN 978-1-4471-5881-3.
48. Lisowski, J. Optimization Methods in Maritime Transport and Logistics. *Pol. Marit. Res.* **2018**, *25*, 30–38. [CrossRef]
49. Lisowski, J. Game Control Methods Comparison when Avoiding Collisions with Multiple Objects Using Radar Remote Sensing. *Remote Sens.* **2020**, *12*, 1573. [CrossRef]
50. Eslami, M. *Theory of Sensitivity in Dynamic Systems*; Springer: Berlin/Heidelberg, Germany, 1994; ebook. [CrossRef]
51. Rosenwasser, E.; Yusupov, R. *Sensitivity of Automatic Control Systems*; CRC Press: Boca Raton, FL, USA, 2019; ebook. [CrossRef]