electronics

MDPI

Article

Synthesis of a Path-Planning Algorithm for Autonomous Robots Moving in a Game Environment during Collision Avoidance

Józef Lisowski

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

Abstract: This paper describes and illustrates the optimization of a safe mobile robot control process in collision situations using the model of a multistep matrix game of many participants in the form of a dual linear programming problem. The synthesis of non-cooperative and cooperative game control software was performed in Matlab/Simulink software to determine the safe path of the robot when passing a greater number of other robots and obstacles. The operation of the game motion control algorithm of a mobile robot is illustrated by computer simulations made in the Matlab/Simulink program of two real previously recorded navigation situations while passing dozens of other autonomous mobile robots.

Keywords: path planning; mobile robot; optimization; safe control; matrix game; computer simulation

Citation: Lisowski, J. Synthesis of a Path-Planning Algorithm for Autonomous Robots Moving in a Game Environment during Collision Avoidance. *Electronics* **2021**, *10*, 675. https://doi.org/10.3390/electronics10060675

Academic Editor: Josep M. Rossell

Received: 23 February 2021
Accepted: 11 March 2021
Published: 13 March 2021

1. Introduction

The range of autonomous mobile robots includes autonomous ground vehicles (AGVs), autonomous aerial vehicles (AAVs), autonomous surface vehicles (ASVs)—in particular, the maritime autonomous surface ship (MASS), described by Munim [1] and Bratic et al. [2]—and autonomous underwater vehicles (AUVs). Path planning, as a fundamental and extensively explored problem in autonomous robotic control, was described by Gwon et al. [3], Kim et al. [4], and Casado and Bermudez [5]. Teso-Fz-Betono et al. [6] showed that effective management of a mobile robot is the most important task of modern navigation. There are many possible solutions to this task, from which the best solution (i.e., the optimal one) is selected, according to the principles presented by Speyer and Jacobson [7] and Yong [8]. Optimization is as effective as the mathematical model of the actual mobile robot control process in collision situations is adequate. Figure 1 illustrates the problem of formulating and solving the optimization task, where the function $Q(s)$ means assessing the quality of the mobile robot control process and adopts the name of the goal control or the quality index control, while s are the decision variables in the form of the strategies of the mobile robot and passing encounters with other robots and obstacles.

According to Bist [9], Bole et al. [10], Statheros et al. [11], and Lei et al. [12], the safety of mobile robots and other autonomous vehicles, such as autonomous ships, depends on the quality of the control processes, both the accuracy of the radar data and the type of model on which the steering algorithm is based.

Thus far, the following methods have been used for designing a guidance system for a mobile robot in the area of concentrated traffic: deterministic methods of static optimization (e.g., linear programming), heuristic methods of static optimization (e.g., ant colony, described by Lazarowska [13]), dynamic optimization methods (e.g., dynamic programming, presented by Lisowski [14]), and methods of artificial intelligence (e.g., neural networks, fuzzy sets, and genetic algorithms, shown by Ahn et al. [15], Stateczny [16], Szlapczynski and Szlapczynska [17], and Rodriguez-Abreo et al. [18]).

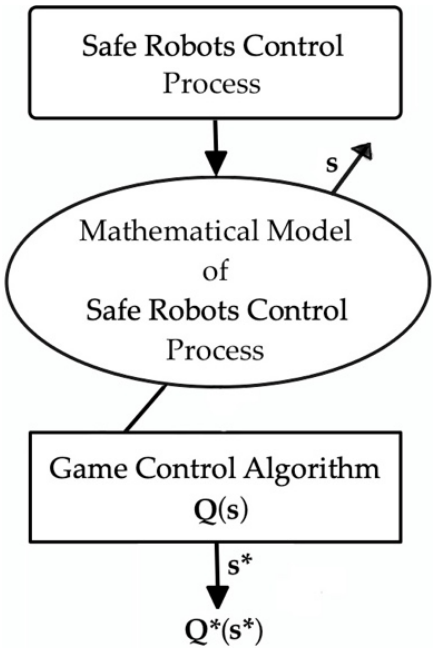

Figure 1. The formulation and solution of the tasks of steering a mobile robot in a game environment: s—permissible strategies; s^*—optimal strategies; Q(s)—quality control index; $Q^*(s^*)$—optimal value of quality control index.

Typically, determining vehicle anticollision strategies is easy, except for in random and troublemaking encounters and in game situations, which are associated with uncertainty of data, the effects of severe weather conditions, and high sea level, as well as uncoordinated actions of other objects, subjectivity in maneuvering decisions, and imprecise recommendations on the nature of the international collision rules, COLREGs. Defining safe strategies is still relevant due to the constantly increasing movement of objects, accompanied by increasing requirements for safe navigation and environmental protection. The largest type of game is about controlling the movement of dynamic objects, including autonomous and nonautonomous marine objects.

The current process of a mobile robot passing other robots and obstacles is characterized by uncertainty and the possibility of an incident in the event of incomplete cooperation between mobile robots. This justifies the problem analysis and method design of safe mobile robot control using game theory methods, such as in the works of Spica et al. [19], Li and Vorobeychik [20], and Liu et al. [21].

The purpose of this study was to synthesize a game algorithm that controls the movement of the mobile robot while passing a larger number of other mobile robots and obstacles, minimizing the risk of collision depending on the scope of cooperation between mobile robots in making maneuvering decisions.

The author's contribution consists of the innovative formulation of the control process of a group of autonomous mobile robots as a multi-person, multistep matrix game, and then the optimization of this control task using dual linear programming.

To this end, in Section 2, the mathematical model of managing a group of mobile robots is presented and the form of the possible robot collision risk index is defined by taking into account weighting factors, depending on the intensity of traffic vehicles.

Then, in Section 3, on the basis of the formulated model, the game control algorithm is synthesized in its three possible versions—as a multistep cooperative game, as a non-cooperative game, and as an optimal non-game control.

Section 4 presents the results of simulation studies of the three algorithms in the navigational passing situation of several autonomous aerial vehicles and in the navigational passing situation of a dozen or so maritime autonomous surface ships.

The results of the research are discussed in Section 5, and detailed conclusions are included in Section 6, indicating the scope of work to be performed in the future.

2. Mathematical Model of Managing a Group of Mobile Robots

The correct model of mobile robot management system process in the situation of k encountered other mobile robots can be mathematically described as a differential game of K participants, as introduced by Isaacs [22]. This mathematically complex description simplifies to a multistep matrix game model, which takes into account the risk of collisions between mobile robots and their possible movement strategies, according to Engwerda [23], Millington and Funge [24], Osborne [25], and Wells [26].

The anticollision radar system ensures automatic monitoring, in which $k = 1, 2, \ldots, K$ encountered mobile robots, showing the quantities describing their movement (speed v_k and course ψ_k) and the parameters of their passing: $d_{k,\min}$ is the shortest approach distance and $t_{k,\min}$ is the time until the critical approach (Figure 2).

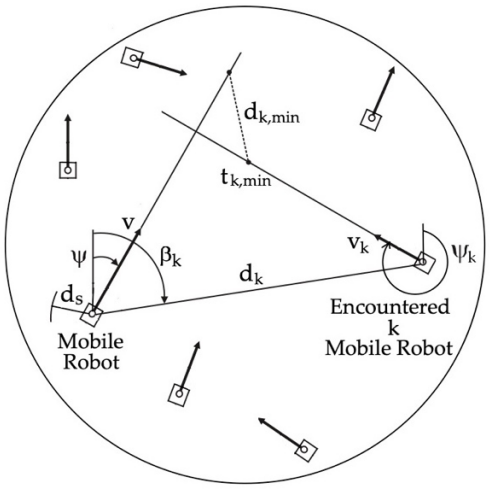

Figure 2. Illustrating the situation of a mobile robot with speed v and course ψ and the met k mobile robot moving with speed v_k and course ψ_k.

Simplifying the dynamic properties of mobile robots until the maneuver is carried out ahead of time allows for the formulation of a matrix game model.

The control variables of the mobile robot are represented by course ψ and speed v, while the control variables for the kth encountered mobile robot are represented by course ψ_k and speed v_k. For the mobile robot, the state variables are represented by risk of collision r_k, while for encountered mobile robot, state variations are represented by distance d_k and bearing β_k (Figure 3).

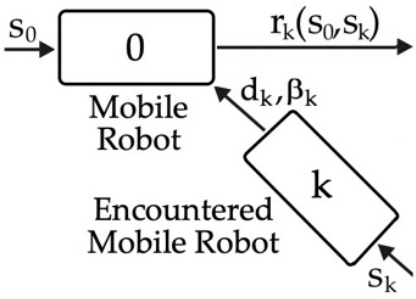

Figure 3. Block representation of a matrix game model of a group of mobile robots: $s_0(\psi_0,v_0)$—mobile robot strategies, $s_k(\psi_k, v_k)$—kth encountered mobile robot strategies, and $r_k(s_0,s_k)$—collision risk of the mobile robot with the kth encountered mobile robot.

Matrix game $\boldsymbol{R}$, in which a player-mobile robot has the ability to use s_0 clean strategies and the player-encountered kth mobile robot has s_k clean strategies, can be described using the following collision risk matrix:

$$\boldsymbol{R}\left(r_k^{s_0,s_k}\right) = \begin{vmatrix} r_1^{1,s_1} & \dots & r_k^{1,s_k} & \dots & r_K^{1,s_K} \\ \dots & & & & \\ r_1^{s_0,s_1} & \dots & r_k^{s_0,s_k} & \dots & r_K^{s_0,s_K} \end{vmatrix} \quad (1)$$

The number of rows in the $\boldsymbol{R}$ matrix is equal to the number of acceptable strategies of mobile robots in the form of maneuvers with the course $\Delta\psi_0$ and speed Δv_0:

$$s_0 = (0,\ \pm\Delta\psi_0,\ \pm 2\Delta\psi_0,\ \dots,\ -\Delta v_0,\ -2\Delta v_0,\ \dots) \quad (2)$$

The number of columns consists of the total number of permissible strategies for all the players involved in a collision situation by analogy with all changes in the course $\Delta\psi_\kappa$ and speed Δv_k of each encountered mobile robot:

$$s_k = (0,\ \pm\Delta\psi_k,\ \pm 2\Delta\psi_k,\ \dots\ -\Delta v_k,\ -2\Delta v_k,\ \dots) \quad (3)$$

Constraints of acceptable strategies (s_0, s_k) are derived from the limitations of the area of operation.

The risk of a collision, $r_k^{s_0,s_k}$, of the mobile robot with a encountered kth other mobile robot can be formulated in the form of the mean square form of the three components of mobile robots approach: the relative shortest approach distance in relation to the previously adopted safe passing distance, $d_{k,\min}$ and d_s, the relative time, $t_{k,\min,}$ of the closest approach to the previously adopted t_s value, and the relative distance, $d_{k,}$ between mobile robots in relation to to the early adopted value for d_s (Figure 4):

$$r_k^{s_0,s_k} = \sqrt{\zeta_d\left(\frac{d_{k,\min}(s_0,s_k)}{d_s}\right)^{-2} + \zeta_t\left(\frac{t_{k,\min}(s_0,s_k)}{t_s}\right)^{-2} + \left(\frac{d_k}{d_s}\right)^{-2}} \quad (4)$$

where ζ_d and ζ_t are the weighting factors, depending on the intensity of vehicle traffic, from 0 to 1.

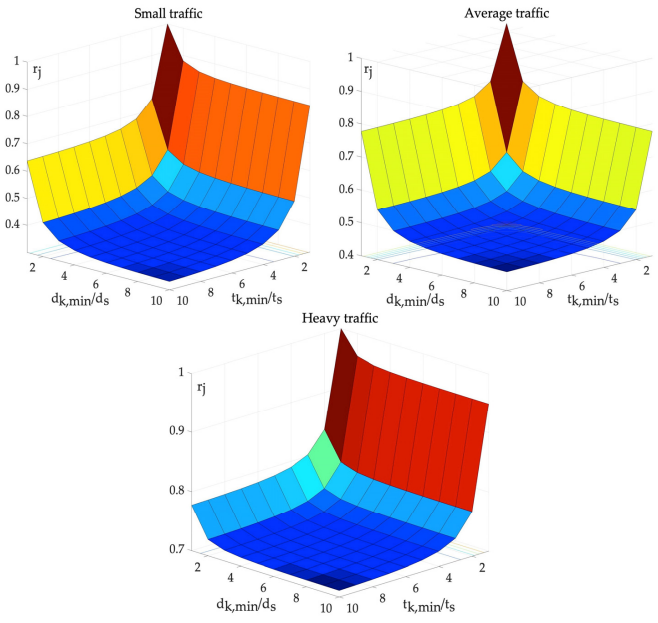

Figure 4. Collision risk surfaces for the three intensities of the traffic situation passing the mobile robot with the kth encountered mobile robot: small traffic—ζ_d= 0.6; ζ_t = 0.3; d_k/d_s = 10; average traffic—ζ_d= 0.4; ζ_t = 0.4; d_k/d_s = 5; heavy traffic—ζ_d= 0.1; ζ_t = 0.4; d_k/d_s = 2.

3. Game Control Algorithm

According to Nisan et al. [27] and Basar and Olsder [28], in most real-world control situations, it is not possible to reach the saddle point in the matrix game and guarantee balance when using pure strategy objects. Therefore, an acceptable solution to the real game can be achieved using a mixed strategy, which represents the probability of implementing the pure player strategy. The probability matrix $\boldsymbol{P}$ of the implementation of pure strategies in the game control of mobile robots takes the following form:

$$\boldsymbol{P}\left(p_k^{s_0,s_k}\right) = \begin{vmatrix} p_1^{1,s_1} & \dots & p_k^{1,s_k} & \dots & p_K^{1,s_K} \\ \dots & & & & \\ p_1^{s_0,s_1} & \dots & p_k^{s_0,s_k} & \dots & p_K^{s_0,s_K} \end{vmatrix} \tag{5}$$

The optimal game control of a mobile robot is the strategy with the highest probability:

$$u_0^* = u_0\left(p_k^{s_0,s_k}\right)_{\max} \tag{6}$$

This optimization problem is convex and can be solved for a small number of permissible strategies using standard Matlab Optimization Toolbox software. In practice, however, it may be necessary to use a greater number of acceptable players strategies, then the solution of such a real-time game control task can be achieved using parallel continuous-time solvers, as presented by Hosseinzadeh et al. [29] for multi-agent smart systems and Nicotra et al. [30] in relation to the problem of optimal control, which can be embedded in the internal states of a dynamic control law running in parallel to the system.

The quality index, Q, of optimal safe control of the mobile robot in the cooperative matrix game takes the following form:

$$Q_c^* = \min_{s_0} \min_{s_k} r_k^{s_0,s_k} \tag{7}$$

In the non-cooperative matrix game, it takes the form of

$$Q_{nc}^* = \min_{s_0} \max_{s_k} r_k^{s_0,s_k} \tag{8}$$

In the usual case of non-game control, the quality index is reduced to the following form:

$$Q_{ng}^* = \min_{s_0} r_k^{s_0,s_k} \tag{9}$$

As a result, by using the principle of dual linear programming, represented by the ***lp*** function in the Optimization Toolbox Matlab/Simulink software, the following algorithms were developed for controlling a group of mobile robots:

- MG_c of a multistep cooperative game for rule (7);
- MG_nc of multistep non-cooperative play for rule (8);
- NG_oc of non-game optimal control for rule (9) (Figure 5).

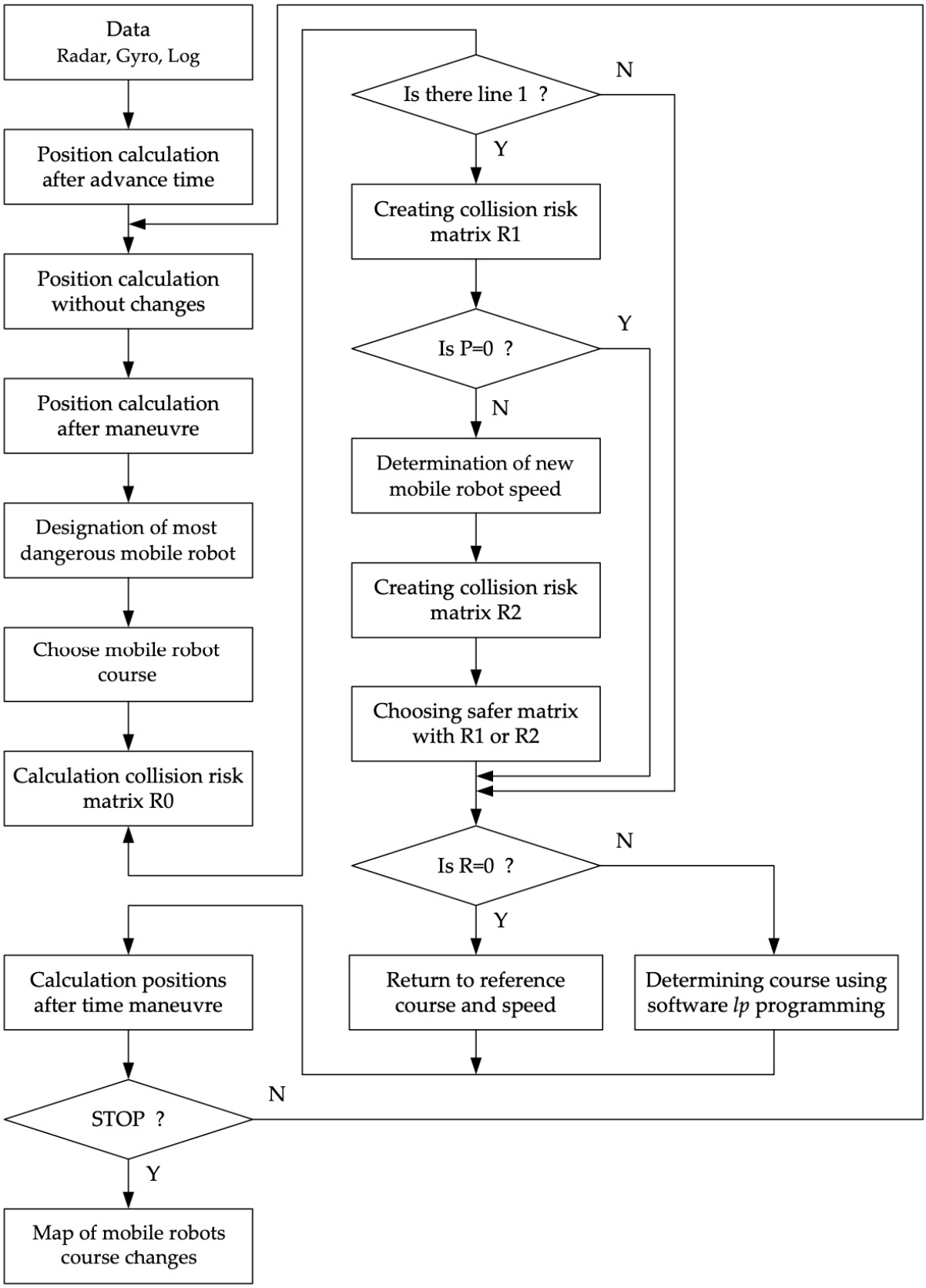

Figure 5. Block diagram of a multistage game control algorithm for a group of mobile robots.

The following (Algorithm 1) is a Matlab/Simulink program for creating a collision risk matrix:

Algorithm 1. Matrix game R

```
Vx = Vo. * sin(Fi) − Vj. * sin(Fij);
Vy = Vo. * cos(Fi) − Vj. * cos(Fij);
Vw = sqrt((Vx.^2) + (Vy.^2));
Dmin = abs((xj. * Vy − yj. * Vx)./abs(Vw));
Tmin = 60 * ((xj. * Vx + yj. * Vy)./(Vw.^2));
q0jj = (Njj − Fi);
id = find((Fi >= (90 * (2 * pi/360)))&(Fi < 180 * (2 * pi/360)));
q0jj(id) = (2 * pi) − Fi(id) + Njj(id);
id = find(q0jj >= (360*(2 * pi/360)));
q0jj(id) = q0jj(id) − (2 * pi);
id = find(q0jj < 0);
q0jj(id) = q0jj(id) + (2 * pi);
Td = 60. * (xj. * sin(Fi) + yj. * cos(Fi))./(Vx. * sin(Fi) + Vy. * cos(Fi));
Dt = (xj. * Vy − yj. * Vx)./(Vx. * sin(Fi) + Vy. * cos(Fi));
BB = Fij − Fi;
id = find(Vw == 0);
Dmin(id) = Dj(id);
id = find(isnan(Tmin));
Tmin(id) = 0;
id = find(Tmin < 0);
Dmin(id) = Dj(id);
Tmin(id) = 0;
wdj = ((Db./Dj).^2) − 0.04;
id = find(wdj < 0);
wdj(id) = 0;
wdm = –0.6213. * log(Dmin./Db) + 1;
id = find(wdm < 0);
wdm(id) = 0;
wtm = ((–Tmin./Tb) + 5)./4;
id = find(wtm < 0);
wtm(id) = 0;
Rj = wdj + wdm. * wtm;
id = find((q0jj >= (90 * (2 * pi/360))) & (q0jj < 270 * (2 * pi/360)));
Rj(id) = 0;
for I = 1:(m * mj);
    for j = 1:n;
        if (Dt(i,j) >= (0.2 * Db) & (BB(i,j) >= 0 * (2 * pi/360) & BB(i,j) <= 270 * (2 * pi/360)));
            Rj(i,j) = 0;
        end
        if (Dt(i,j) < (–0.2 * Db) & (BB(i,j >= 90 * (2 * pi/360) & BB(i,j) <= 360 * (2 * pi/360)));
            Rj(i,j) = 0;
        end
    end
end
if cala == 0;
id = find(Rj > 1);
Rj(id) = 1;
end
if (max(sum(Rj')) == n);
cala = 1;
end
```

4. Results

The developed path-planning algorithms of autonomous mobile robots were subjected to a computer simulation on two examples of real navigation situations of groups of mobile robots—autonomous aerial vehicles (AAV) and maritime autonomous surface ships (MASS).

The simulation studies included a different number of autonomous mobile robots: $K = 3$ and $K = 17$. Moreover, the calculations of the safe path of vehicles were carried out for different values of the safe distance: $d_s = 4$ m and $d_s = 12$ m or $d_s = 0.5$ nm and $d_s = 1.5$ nm, corresponding to good and restricted weather conditions.

4.1. Computer Simulation of the Navigational Situation of Autonomous Aerial Vehicles Traffic

The simulation of MG_c, MG_nc, and NG_oc computer programs was made in Matlab/Simulink in a small-traffic situation of the mobile robot passing $K = 3$ other encountered mobile robots in good weather conditions ($d_s = 4$ m) and in unfavorable weather conditions ($d_s = 12$ m) (Table 1 and Figures 6–8).

Table 1. Data of mobile robot 0 and encountered (K = 3) mobile robots.

Mobile Robot	Distance d_k (m)	Bearing β_k (°)	Speed v_k (m/s)	Course ψ_k (°)
0	-	-	20.0	0
1	88	326	14.5	90
2	143	6	16.2	180
3	75	11	16.0	200

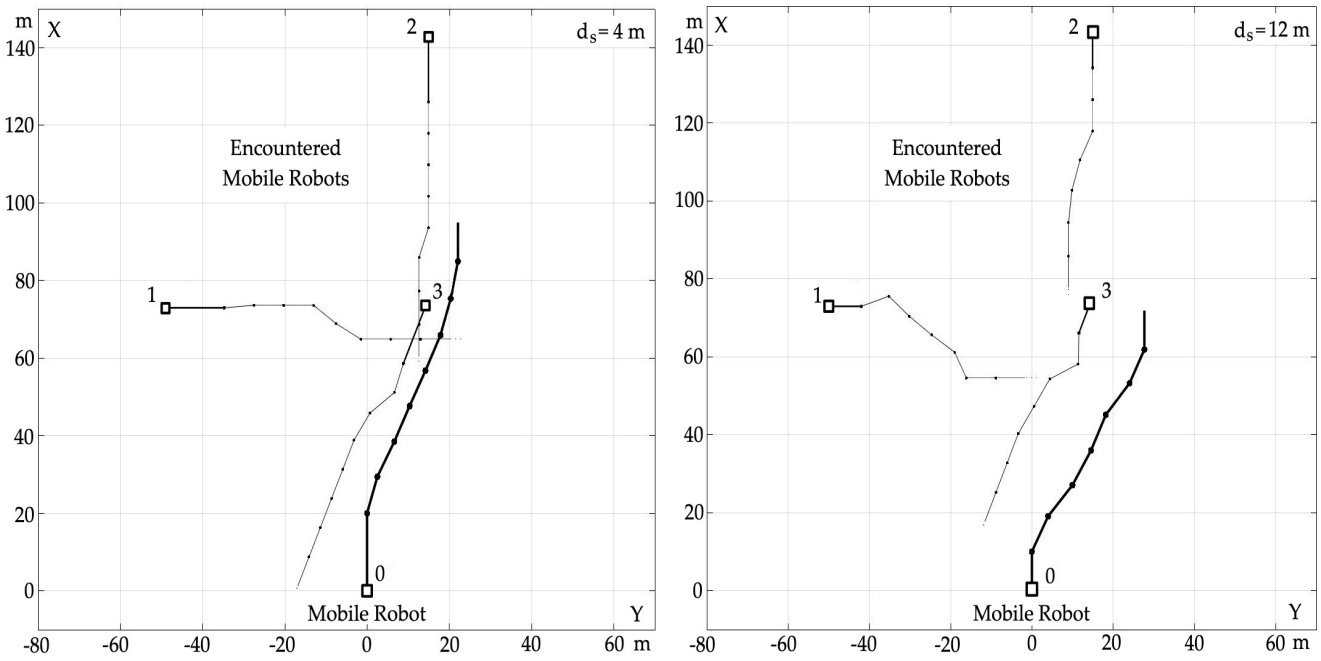

Figure 6. Mobile robot safe path when passing $K = 3$ encountered mobile robots in good weather conditions ($d_s = 4$ m) and in unfavorable weather conditions ($d_s = 12$ m), determined by the MG_c algorithm of the cooperative matrix game.

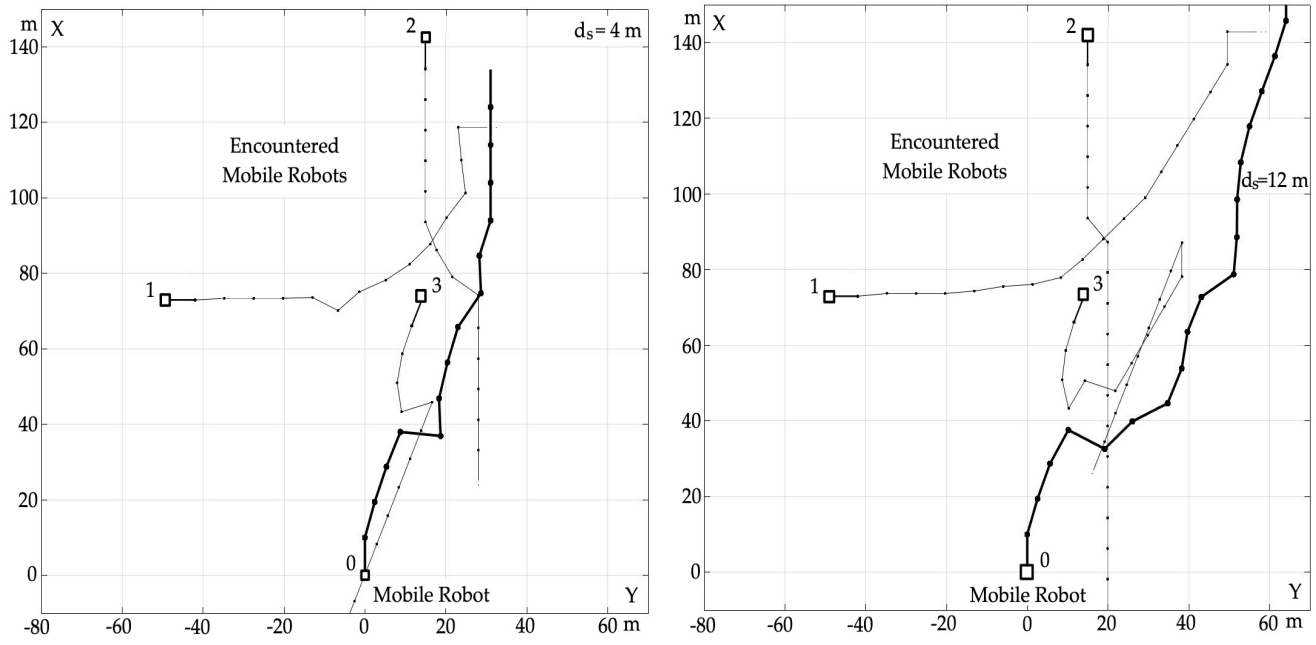

Figure 7. Mobile robot safe path when passing $K = 3$ encountered mobile robots in good weather conditions ($d_s = 4$ m) and in unfavorable weather conditions ($d_s = 12$ m), determined by the MG_nc algorithm of the non-cooperative matrix game.

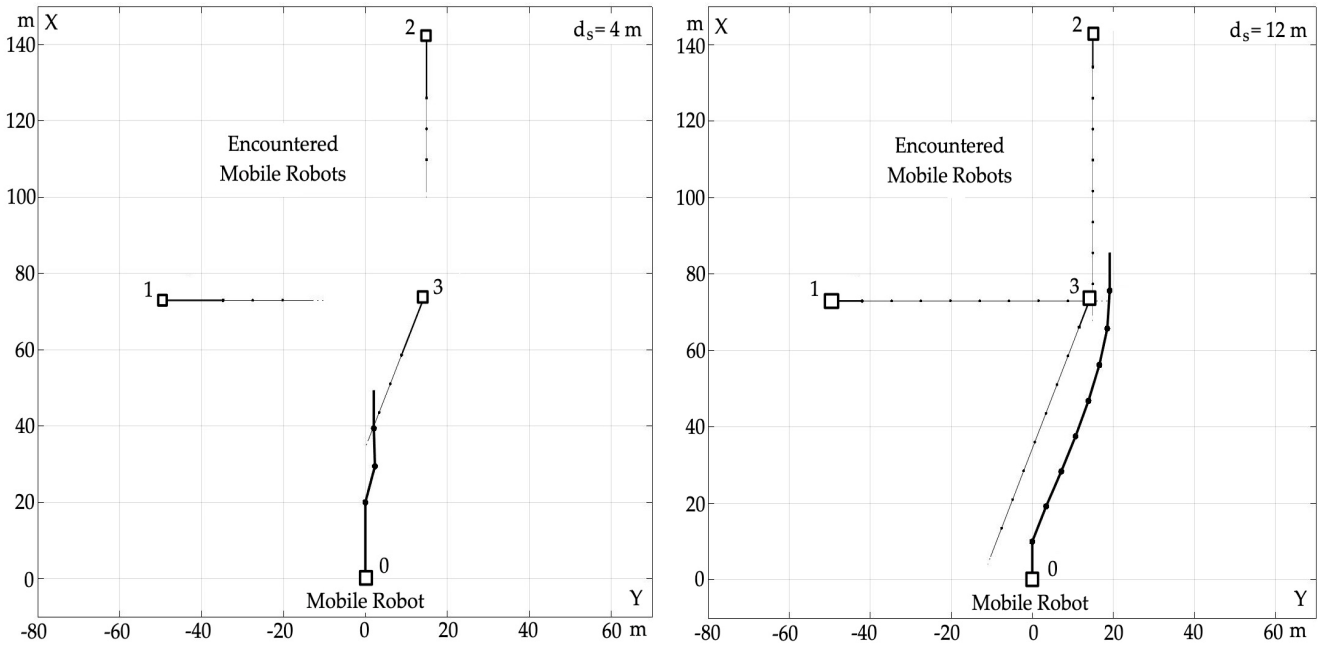

Figure 8. Mobile robot safe path when passing $K = 3$ encountered mobile robots in good weather conditions ($d_s = 4$ m) and in unfavorable weather conditions ($d_s = 12$ m), determined by the NG_oc algorithm of the non-game control.

Figure 9 shows a comparison of the trajectory of a mobile robot while passing other mobile robots encountered, determined according to three control methods, in good and unfavorable weather conditions.

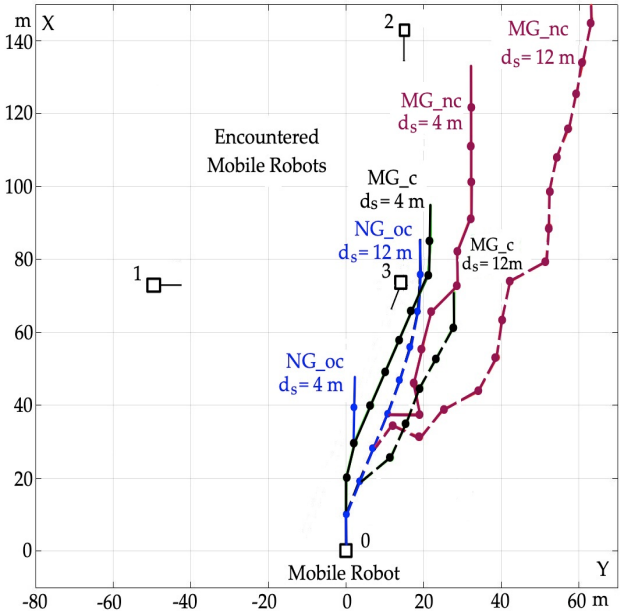

Figure 9. Comparison of safe mobile robot paths when passing $K = 3$ encountered mobile robots in good weather conditions ($d_s = 4$ m) and unfavorable weather conditions ($d_s = 12$ m), determined by the MG_c, MG_nc and NG_oc algorithms.

4.2. Computer Simulation of Navigational Situation of Maritime Autonomous Surface Ships Traffic

The simulation of MG_c, MG_nc, and NG_oc computer programs was made in Matlab/Simulink in the situation a heavy traffic of the maritime autonomous surface ship passing $K = 17$ other encountered autonomous surface ships in good ($d_s = 0.5$ nm) and restricted ($d_s = 1.5$ nm) visibility at sea (Table 2 and Figures 10–12).

Table 2. Kinematic data of the autonomous surface vessel 0 and the encountered ($K = 17$) autonomous surface ships.

Autonomous Surface Ship	Distance d_k (nm)	Bearing β_k (°)	Speed v_k (kn)	Course ψ_k (°)
0	-	-	13	100
1	4	140	3	350
2	9	120	4	330
3	3	47	5	220
4	1	200	5	100
5	4	105	5	120
6	5	140	7	58
7	6	48	5	150
8	5	85	1	150
9	10	120	3	20
10	4	140	2	350
11	6	140	4	350
12	8	65	10	205
13	4	155	5	50
14	3	20	8	140
15	8	145	7	0
16	5	10	15	150
17	4	300	10	100

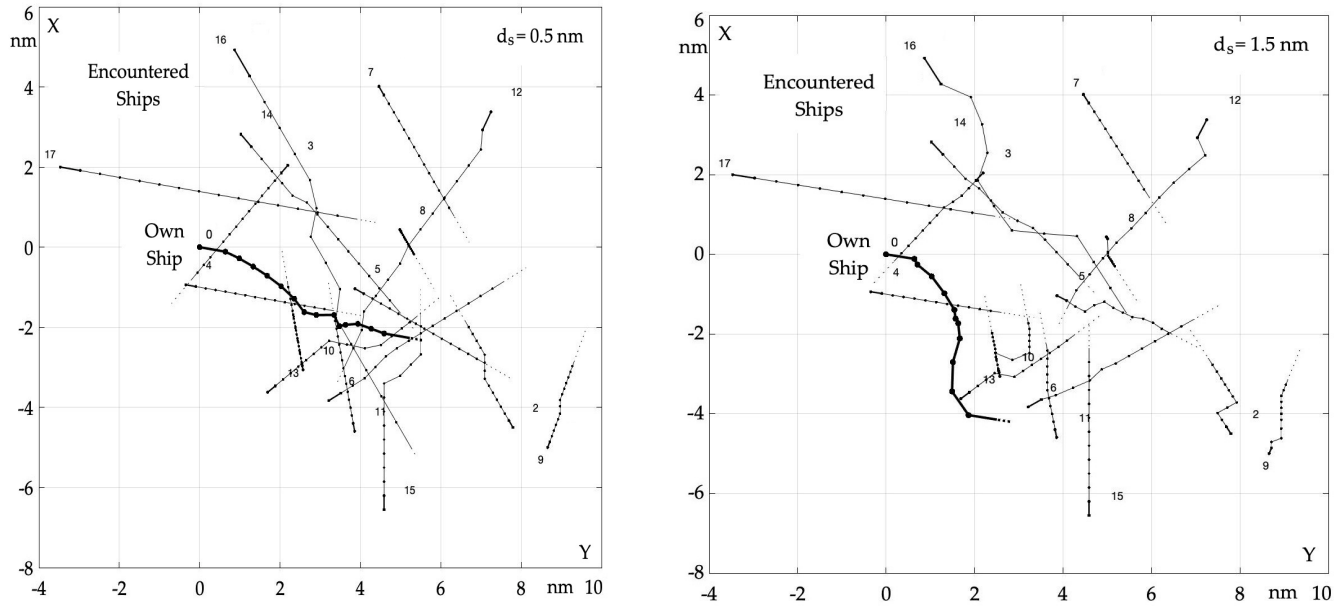

Figure 10. Autonomous surface ship safe path when passing K = 17 encountered autonomous surface ships in good (d_S = 0.5 nm) and restricted (d_S = 1.5 nm) visibility at sea, determined by MG_c algorithm of cooperative matrix game.

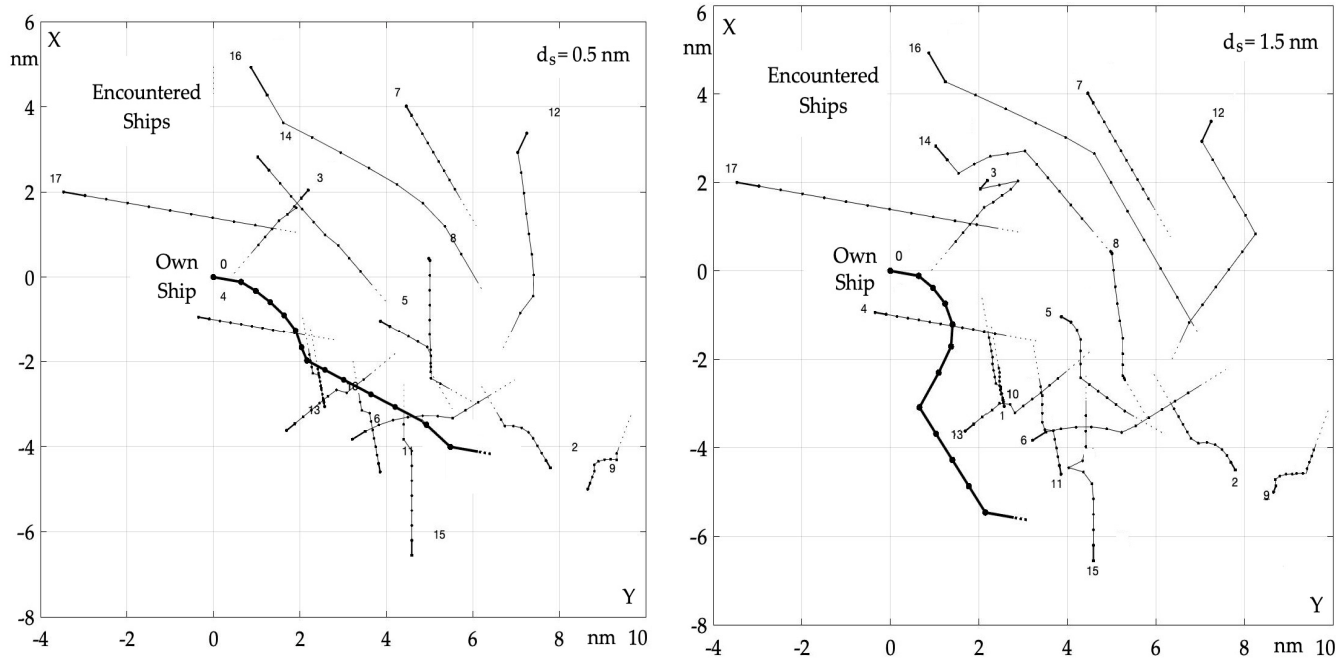

Figure 11. Autonomous surface ship safe path when passing K = 17 encountered other autonomous surface ships in good (d_S = 0.5 nm) and restricted (d_S = 1.5 nm) visibility at sea, determined by MG_n algorithm of non-cooperative matrix game.

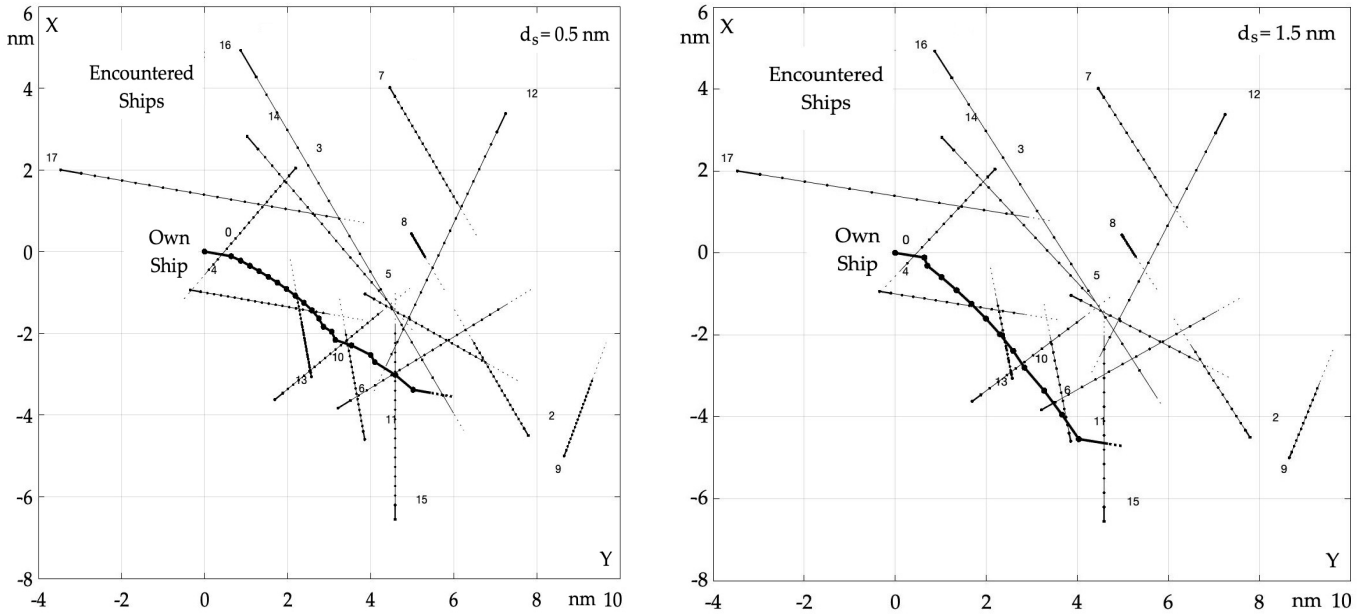

Figure 12. Autonomous surface ship safe path when passing K = 17 encountered other autonomous surface ships in good (d_s = 0.5 nm) and restricted (d_s = 1.5 nm) visibility at sea, determined by NG_oc algorithm of non-game control.

Figure 13 shows a comparison of the autonomous surface ship safe path when passing other encountered autonomous surface ships, determined in good and restricted visibility at sea, according to three controlling programs. The measure of the control quality is the deflection of the autonomous surface ship's path from the initial direction of movement. Greatest deflection occurs with non-cooperative movement of autonomous surface ships, and smaller deflection occurs with cooperation of ships. On the other hand, with non-game control, encountered autonomous surface ships maintain their course and speed; they pass each other optimally at a previously set safe distance, but with higher risk of collision.

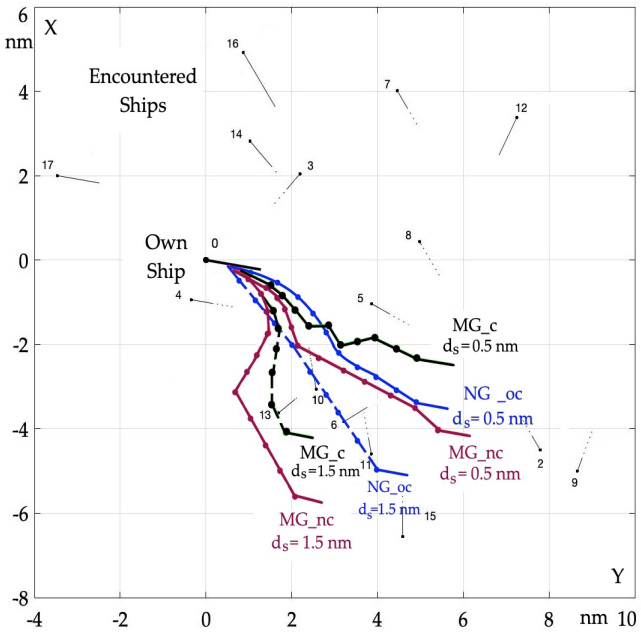

Figure 13. Comparison of autonomous surface ship safe paths when passing K = 17 encountered other autonomous surface ships in good (d_s = 0.5 nm) and restricted (d_s = 1.5 nm) visibility at sea, determined by the MG_c, MG_nc, and NG_oc algorithms.

5. Discussion

The presented synthesis and computer simulation of the path-planning algorithms of autonomous mobile robots, taking into account the conditions of optimal and game control, in terms of cooperation or lack of cooperation between autonomous mobile robots, demonstrate a more reliable and safe solution to the task of controlling this process.

The measure of solving the control task is the final deviation of the mobile robot's trajectory from the initial direction of movement. The game ended at the moment t_k, when the risk of own ship, $r_{k,}$ in relation to each ship, k, reached the value of zero ($r_k(t_k) = 0$). Then, the final deflection of the trajectory of the own ship from the reference trajectory was assessed. The greatest deviation occurs with non-cooperative movement of objects, and smaller deviation occurs with cooperation of objects. On the other hand, with non-game control, the encountered mobile robots maintain their course and speed; they pass each other optimally at a previously set safe distance, but with a higher risk of collision.

An important factor influencing the course of the safe trajectory of an autonomous mobile robot is the environmental conditions. In the case of autonomous aerial robots, it is the density of vehicle traffic, and in the case of marine autonomous surface ships, it is the visibility conditions at sea.

Additionally, the control algorithm must be equipped with a procedure for semantic interpretation of the legal rules of anticollision maneuvering COLREGs, depending on the state of visibility at sea.

6. Conclusions

The application of a model of cooperative and non-cooperative multistep matrix games for designing control programs enables the determination of an optimal and safe path for an autonomous mobile robot in situations where it passes a greater number of other autonomous mobile robots.

The path planning of an autonomous surface ship was treated as a combination of course and speed change maneuvers of autonomous surface ships.

The synthesis of the game control programs takes into account the degree of cooperation in maneuvering decisions between autonomous surface ships, the time of the advance maneuver, and the path deflection from the given direction of movement.

The final deflection of the current path from the reference path significantly depends on the cooperation of the mobile robot with other encountered robots.

In following studies, the use of selected methods of computational intelligence can be analyzed; in particular, fuzzy control, allowing for better adjustment of control to the environmental conditions, and neural network, allowing for more accurate mapping of vehicles encountered as mobile obstacles.

The potential application of the research presented in this article can be especially indicated in the improvement of maritime autonomous surface ship (MASS) control software.

Funding: This research was funded by a research project of the Electrical Engineering Faculty, Gdynia Maritime University, Poland, No. WE/2021/PZ/02: "Control theory and artificial intelligence techniques in optimal and safe ship operation".

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. Munim, Z.H. Autonomous ships: A review, innovative applications and future maritime business models. *Supply Chain Forum Int. J.* **2019**, *20*, 266–279. [CrossRef]
2. Bratić, K.; Pavić, I.; Vukša, S.; Stazić, L. Review of Autonomous and Remotely Controlled Ships in Maritime Sector. *Trans. Marit. Sci.* **2019**, *8*, 253–265. [CrossRef]
3. Gwon, J.; Kim, H.; Bae, H.; Lee, S. Path Planning of a Sweeping Robot Based on Path Estimation of a Curling Stone Using Sensor Fusion. *Electronics* **2020**, *9*, 457. [CrossRef]
4. Kim, C.; Kim, Y.; Yi, H. Fuzzy Analytic Hierarchy Process-Based Mobile Robot Path Planning. *Electronics* **2020**, *9*, 290. [CrossRef]

5. Casado, R.; Bermúdez, A. A Simulation Framework for Developing Autonomous Drone Navigation Systems. *Electronics* **2020**, *10*, 7. [CrossRef]
6. Teso-Fz-Betoño, D.; Zulueta, E.; Fernandez-Gamiz, U.; Aramendia, I.; Uriarte, I. A Free Navigation of an AGV to a Non-Static Target with Obstacle Avoidance. *Electronics* **2019**, *8*, 159. [CrossRef]
7. Speyer, J.L.; Jacobson, D.H. *Primer on Optimal Control Theory*; Siam: Philadelphia, PA USA, 2010; ISBN 978-0-898716-94-8.
8. Yong, J. *Optimization Theory—A Concise Introduction*; Word Scientific: Singapore, 2018; ISBN 978-981-3237-64-3.
9. Bist, D.S. *Safety and Security at Sea*; Butter Heinemann: Berlin, Germany, 2000; ISBN 0-75064-774-4.
10. Bole, A.; Dineley, B.; Wall, A. *Radar and ARPA Manual*; Elsevier: Amsterdam, The Netherlands, 2006; ISBN 978-0-08-048052-7.
11. Statheros, T.; Howells, G.; Maier, K.M. Autonomous Ship Collision Avoidance Navigation Concepts, Technologies and Techniques. *J. Navig.* **2007**, *61*, 129–142. [CrossRef]
12. Lei, X.; Feng, B.; Wang, G.; Liu, W.; Yang, Y. A Novel FastSLAM Framework Based on 2D Lidar for Autonomous Mobile Robot. *Electronics* **2020**, *9*, 695. [CrossRef]
13. Lazarowska, A. Safe Ship Trajectory Planning Based on the Ant Algorithm—The Development of the Method. *Act. Navig.* **2015**, 153–160. [CrossRef]
14. Lisowski, J. Multistage Dynamic Optimization with Different Forms of Neural-State Constraints to Avoid Many Object Collisions Based on Radar Remote Sensing. *Remote Sens.* **2020**, *12*, 1020. [CrossRef]
15. Ahn, J.-H.; Rhee, K.-P.; You, Y.-J. A study on the collision avoidance of a ship using neural networks and fuzzy logic. *Appl. Ocean Res.* **2012**, *37*, 162–173. [CrossRef]
16. Stateczny, A. Neural maneuvre detection of the tracked target in ARPA system. In Proceedings of the IFAC Conference on Control Applications in Marine Systems Location, Glasgow, UK, 18–20 July 2001; University Strathclyde: Glasgow, UK, 2001; pp. 209–214.
17. Szlapczynski, R.; Szlapczynska, J. A method of determining and visualizing safe motion parameters of a ships navigating in restricted waters. *Ocean Eng.* **2016**, *129*, 363–373. [CrossRef]
18. Rodríguez-Abreo, O.; Garcia-Guendulain, J.M.; Hernández-Alvarado, R.; Rangel, A.F.; Fuentes-Silva, C. Genetic Algorithm-Based Tuning of Backstepping Controller for a Quadrotor-Type Unmanned Aerial Vehicle. *Electronics* **2020**, *9*, 1735. [CrossRef]
19. Spica, R.; Cristofalo, E.; Wang, Z.; Montijano, E.; Schwager, M. A Real-Time Game Theoretic Planner for Autonomous Two-Player Drone Racing. *IEEE Trans. Robot.* **2020**, *36*, 1389–1403. [CrossRef]
20. Li, Y.; Vorobeychik, Y. Path planning games. *arXiv* **2019**, arXiv:1910.13880.
21. Liu, Z.; Wu, Z.; Zheng, Z. A cooperative game approach for assessing the collision risk in multi-vessel encountering. *Ocean Eng.* **2019**, *187*, 106175. [CrossRef]
22. Hermes, H.; Isaacs, R. Differential Games. *Math. Comput.* **1965**, *19*, 700. [CrossRef]
23. Engwerda, J.C. *LQ Dynamic Optimization and Differential Games*; John Wiley & Sons: Hoboken, NJ, USA, 2005; ISBN 978-0-470-01524-7.
24. Millington, I.; Funge, J. *Artificial Intelligence for Games*; CRC Press: Boca Raton, FL, USA, 2018.
25. Osborne, M.J. *An Introduction to Game Theory*; Oxford University Press: New York, NY, USA, 2004; ISBN 978-0-19-512895-6.
26. Wells, D. *Game and Mathematics*; Cambridge University Press: London, UK, 2003; ISBN 978-1-78326-752-1.
27. Nisan, N.; Roughgarden, T.; Tardos, E.; Vazirani, V.V. *Algorithmic Game Theory*; Cambridge University Press: New York, NY, USA, 2007; ISBN 978-0-521-87282-9.
28. Basar, T.; Olsder, G.J. *Dynamic Non-Cooperative Game Theory*; Siam: Philadelphia, PA, USA, 2013; ISBN 978-0-898-714-29-6.
29. Hosseinzadeh, M.; Garone, E.; Schenato, L. A Distributed Method for Linear Programming Problems with Box Constraints and Time-Varying Inequalities. *IEEE Control. Syst. Lett.* **2018**, *3*, 404–409. [CrossRef]
30. Nicotra, M.M.; Liao-McPherson, D.; Kolmanovsky, I.V. Embedding Constrained Model Predictive Control in a Continuous-Time Dynamic Feedback. *IEEE Trans. Autom. Control.* **2018**, *64*, 1932–1946. [CrossRef]